RSA Incident Response: Threat Detection Techniques - Point of Sale Attacks

RSA Security

January 2014

Table of Contents

1. Executive Summary

The recent surge in news stories, white papers, blog posts, interviews, and technical briefings regarding the Point of Sale (POS) system breaches at many major retailers, has left most organizations speculating as to whether or not they could be susceptible to the same type of attack. Many security companies are claiming that they can protect organizations against this type of attack, some even claiming that this incident was sophisticated and advanced. RSA IR analyzed many of the samples that were used in the attack against the largest of these companies and based on our analysis; the actual malware that was used on the POS endpoints appears to have been in the wild since at least June of 2013. Some organizations are trying to treat this threat symptomatically. Instead, RSA suggests that organizations should look at how an intruder would get from outside of the network to POS machines and what measures are in place for detection and identification for this type of intrusion.

The malware that was used in this breach has been well documented by many research companies, some of which have claimed to attribute an author to the different pieces of malware. This report will not delve into the technical artifacts of the malware, but simply how RSA tools like Security Analytics and ECAT would have alerted an organization about this type of intrusion, leading to expedited response time, reduced exposure, and subsequently helping stop the attack before data was exfiltrated. Included along with this report is content that can be deployed to RSA products to detect different aspects of this attack.

The accompanying digital appendix includes Yara Signatures that can be used by organization to determine if they currently have these types of malicious files present in their enterprise. Also available in the digital appendix is a Blacklist that can be imported into ECAT to help an organization quickly identify and categorize known files.

2. ECAT Detection

One of the types of malware that was used in the attacks was a tool that captures memory from processes and parses the data for credit card track data. During testing, ECAT was able to detect this malware, even before RSA IR created specific content for this variant of memory scraping malware. Figure 1 illustrates that ECAT detected an unknown process, providing context that the activity is related to the "POSWDS" service running on the machine.

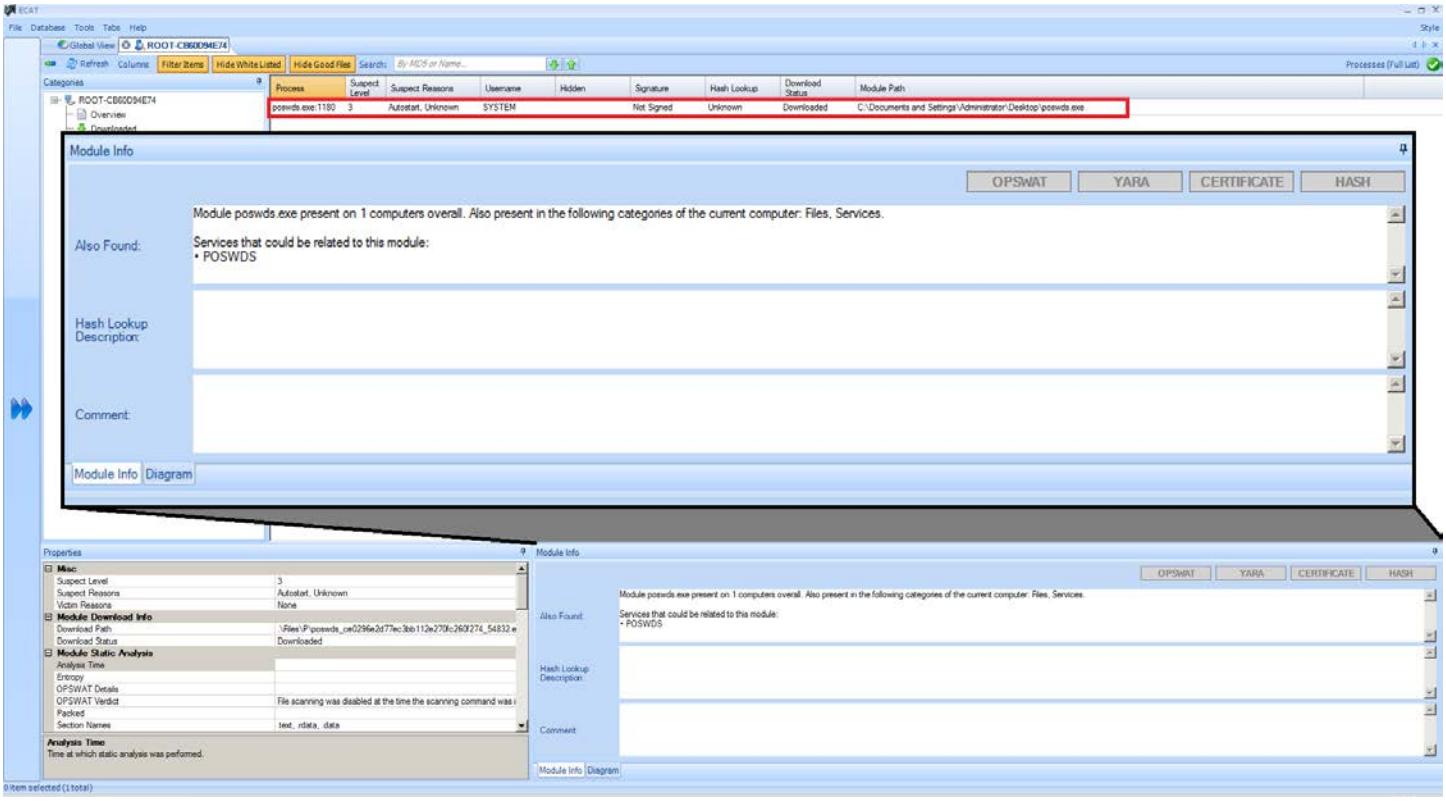

Figure 1: ECAT malicious process detection

Figure 2 highlights the malicious service that was started by the BlackPOS malware. Using ECAT to conduct daily interval scanning to detect malicious processes and services will allow the ECAT administrator to review the Module Info box, highlighted in Figure 1 to see how many machines are running the process.

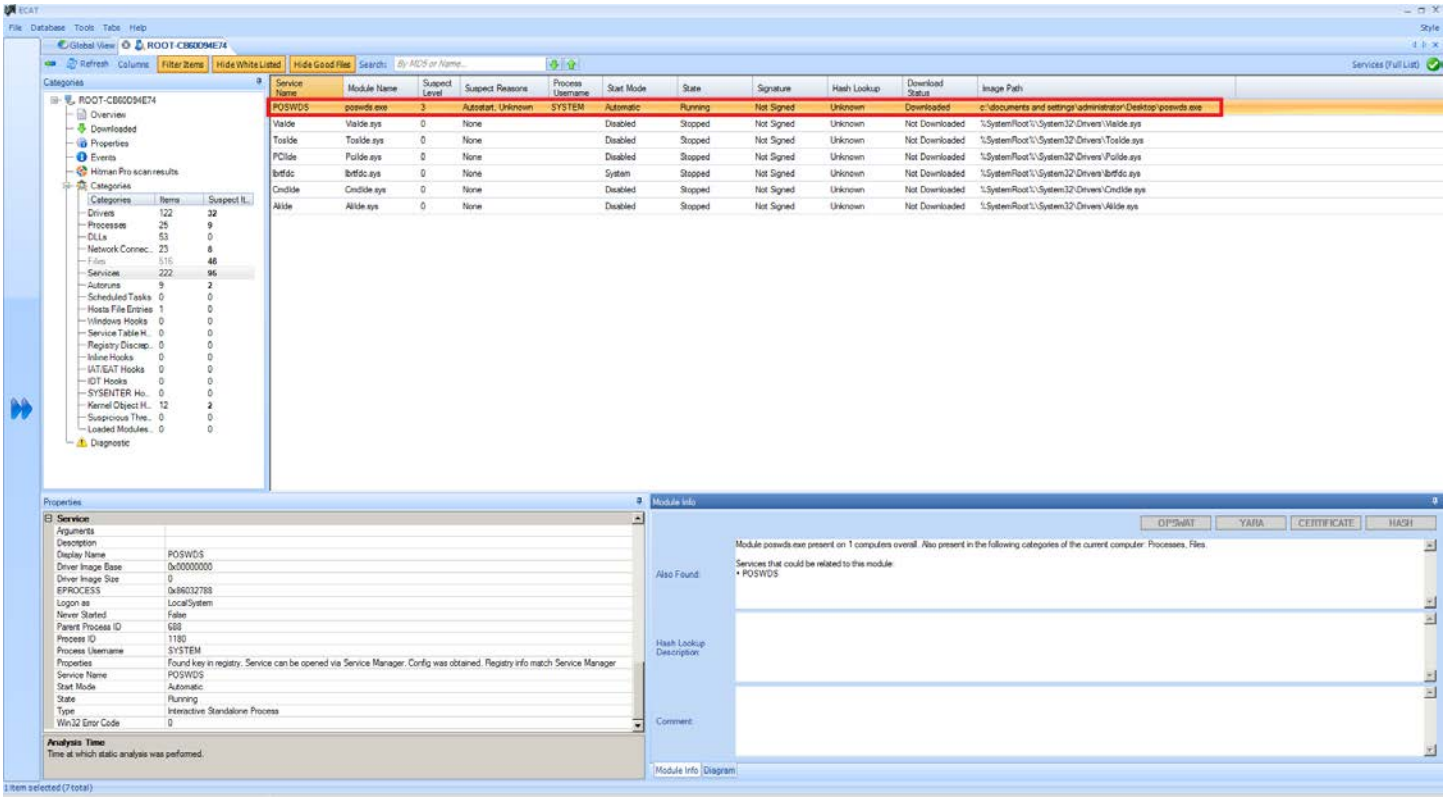

Figure 2: ECAT detection of malicious service

Figure 3 further demonstrates ECAT's inherent ability to detect and identify files that can be used maliciously in environments. When the comparison fails, it reports the process in memory with an "image mismatch" as shown under the column entitled "Suspect Reasons". "Image mismatch" means the binary has a custom code loading method such as a packer. ECAT compares the code sections from the file on disk to the code sections for the process in memory. If the sections differ from each other greatly, the process in memory is marked as "image mismatch." Any other differences between the two will be highlighted by ECAT as hooks or code injection.

Packers allow malware authors to use the same code over and over again while ensuring that the binary has different MD5 hashes on disk. Some packers are used to thwart malware reversing efforts to understand the code. After the binary is loaded into memory, the packer is the first part of the executable that, when executed, will decompresses and/or decrypt the rest of the code. After this is done, the normal execution of the code occurs. The file depicted in this figure was a Netcat variant that the author used a custom packer and encryption to hide the real binary.

File Name	Suspect Level	Suspect Reasons	Hidden	Signature	Hash Lookup	Download Status	File Path
[MEM HASH] malware.exe	52	Image Mismatch, Malware Found		Not Signed	Unknown	Downloaded	C:\Documents and Settings\Administrator\Desktop\malware.exe
malware.exe	39	Network Access, Malware Found, ...		Not Signed: Рабочая группа Twain	Unknown	Downloaded	C:\Documents and Settings\Administrator\Desktop\malware.exe

Figure 3: ECAT detection of tools being used maliciously

ECAT can also track network processes connections. Figure 4 shows ECAT tracking Netcat network connections between machines and what ports were used. Any abnormal workstation to workstation communications should be investigated.

Process Name	Module Name	Suspect Level	Suspect Reasons	Port	Address	State	Connection Analysis	Proxy Address	Signature	Hash Lookup	Download Status	Doma Host Name
netcat.exe	netcat.exe	2	Network Access	80	172.16.13.12	Statistic	None	None	Not Signed	Unknown	Not Downloaded	[redacted]
netcat.exe	netcat.exe	2	Network Access	80	172.16.13.12	Statistic	None	None	Not Signed	Unknown	Not Downloaded	[redacted]
netcat.exe	netcat.exe	2	Network Access	8080	172.16.13.12	Statistic	None	None	Not Signed	Unknown	Not Downloaded	[redacted]
netcat.exe	netcat.exe	2	Network Access	31337	172.16.13.135	Statistic	None	None	Not Signed	Unknown	Not Downloaded	root-...

Figure 4: ECAT tracking network connections

If the attacker uses an executable to schedule a task, an analyst can review the system for scheduled tasks using ECAT. Figure 5 shows an example of a scheduled backdoor detected with ECAT.

Module Name	Suspect Level	Suspect Reasons	Next Run	Last Run	Signature	Hash Lookup	Download Status	Module Path
netcat.exe	5	Autostart, Attribute Hidden, Unknown	2/3/2014 6:15:00 PM	2/2/2014 6:15:00 PM	Not Signed: Рабочая группа Twain	Unknown	Downloaded	c:\windows\system32\netcat.exe

Figure 5: ECAT detection of scheduled backdoors

Finally, using ECAT's Global Module List, an analyst can quickly review the data collected by ECAT across all the machines. Using the built in filtering options, the malicious Netcat process bubbles to the top as shown in Figure 6. Sometimes attackers will turn on the hidden file attribute to hide their tools as shown under Suspect Reasons as "Attribute Hidden".

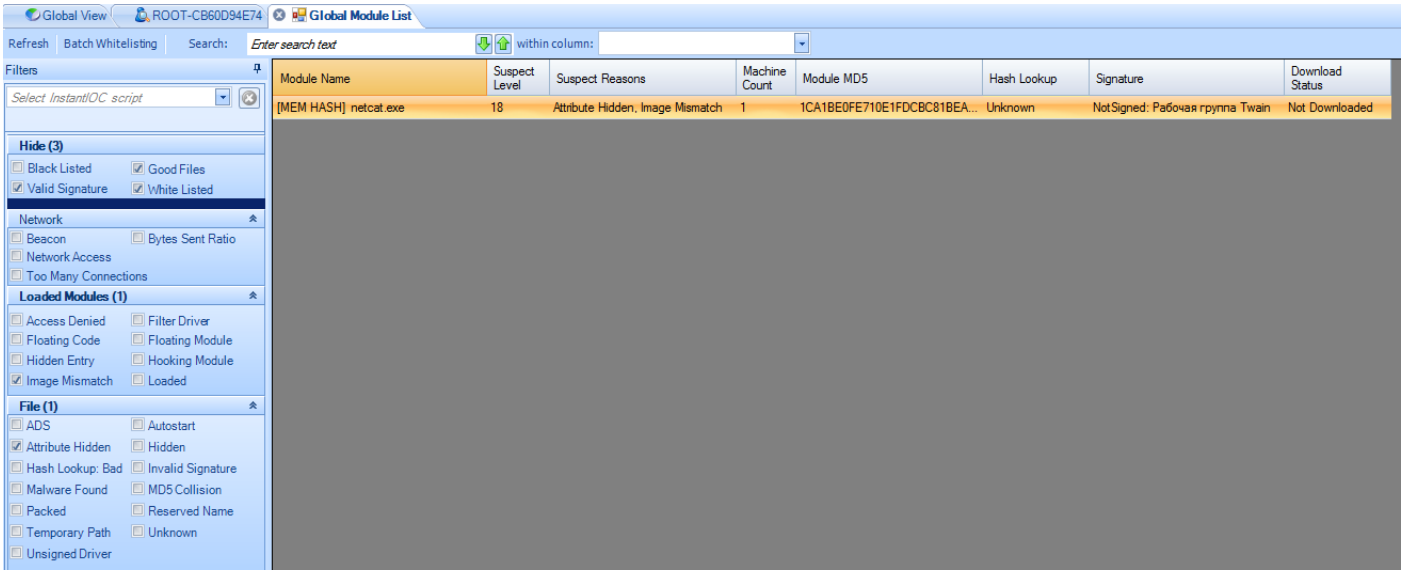

Figure 6: ECAT's Global Module List

3. Security Analytics Detection

3.1 Malware Detection

Security Analytics Malware Analysis engine is an automated malware analysis processor designed to analyze certain types of file objects (for example, Windows PE, PDF, and MS Office) to assess the likelihood that a file is malicious. Using the Malware Analysis engine, an analyst can prioritize the massive number of files captured in order to focus efforts on the files that are most likely to be malicious.

Security Analytics Malware Analysis detects indicators of compromise using four distinct analysis methodologies:

- Network Session Analysis (network)
- Static File Analysis (static)
- Dynamic File Analysis (sandbox)
- Security Community Analysis (community)

Each of the four distinct analysis methodologies is designed to compensate for inherent weaknesses in the others. For example, Dynamic File Analysis can compensate for Zero-Day attacks that are not detected during the Security Community Analysis phase. By avoiding malware analysis that strictly focuses on one methodology, the analyst is more likely to be shielded from false negative results.

Configuration of the Malware Engine is extremely flexible allowing for the selective use of community and external Sandbox services to help limit exposure of sensitive information in certain situations. Figure 7 shows the initial summary of analysis performed by the Malware Engine. In this example, the analyst did not submit to the community or use an AV engine to analyze the malware.

File Summary

Overall	Static	Community	Sandbox	AV	File Name	File Hash
		n/a			853fb5a2aad2e0533e390cfa5b0f3dfe96a054390cacdc8f4ba844bba20809e4.exe	4d445b11f9cc3334a4925a7ae5ebb2b7

Figure 7: Security Analytics File Summary

Based on the analysis performed by the Malware Engine, it can be easily determined that this file is malicious. Figure 8 below shows the Top 10 Indicators of Compromise from the static analysis.

Top 10 Event Indicators of Compromise

	Static (PE) - Meta: Stripped of Informational Meta Strings File: 853fb5a2aad2e0533e390cfa5b0f3dfe96a054390cacdc8f4ba844bba20809e4.exe, type: IMAGE_FILE_MACHINE_I386, size: 98304, pe size: 98304, md5: 4d445b11f9cc3334a4925a7ae5ebb2b7, sha1: e0269fe0f965f5d629b3caf04c088a8e16a4e3d2
	Static (PE) - Artifact: Security Weakening - cmd.exe usage Yara rule: WeakenCmdExe in file: rsa_mw_pe_artifacts.yara has detected a malicious string: cmd.exe at offset: 82768
	Sandbox - IOC: Indicator of Compromise - High Severity IOC (ioc: process-injected-thread, name: Process Injected and Launched Thread In Another Process, severity: 95, confidence: 90)
	Static (PE) - File Size: Abnormally Small in Size (<100k) File: 853fb5a2aad2e0533e390cfa5b0f3dfe96a054390cacdc8f4ba844bba20809e4.exe, type: IMAGE_FILE_MACHINE_I386, size: 98304, pe size: 98304, md5: 4d445b11f9cc3334a4925a7ae5ebb2b7, sha1: e0269fe0f965f5d629b3caf04c088a8e16a4e3d2
	Static (PE) - DLL Imports: Contains Watch-Listed Fingerprint(s) Import Function Name: createservicea
	Sandbox - IOC: Indicator of Compromise - Medium Severity IOC (ioc: cmd-exe-file-execution, name: Command Exe File Execution Detected, severity: 50, confidence: 80)
	Static (PE) - Packers: Packer Signature Matches Armadillo Yara rule: Armadillo in file: rsa_mw_pe_packers.yara has detected a malicious string: Ux8bxecjxffhxa0CAx00hxc4xb0@x00dxa1 at offset: 14384
	Static (PE) - Artifact: Reconnaissance Activity (Inventory Active Services) Yara rule: ReconServices in file: rsa_mw_pe_artifacts.yara has detected a malicious string: net start at offset: 82319
	Static (PE) - Checksum: No Checksum Value CheckSum Value Set to: 0x0

Figure 8: Top 10 Event Indicators of Compromise – Static Analysis

Finally, additional value was gained from dynamic analysis of the file through execution in a sandbox. Figure 9 shows that this file performs process injection to ensure that this file is run when another process is loaded.

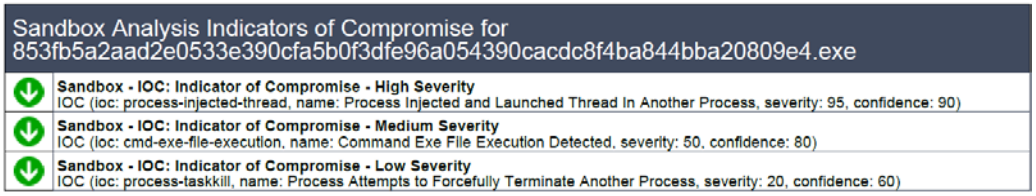

Figure 9: Top 10 Indicators of Compromise – Sandbox

3.2 Lateral Movement Techniques

In POS system breaches, attackers often penetrate the network from vulnerable web facing servers, navigating through the network, until they are able to reach the POS systems. This type of lateral movement can easily be detected using Security Analytics before the attackers even reach their intended targets. Using the Windows "net" commands, scheduling tasks using the AT command, and using freely available system administration tools like PsExec are common practices that are successfully employed by attackers. This type of traffic is easily recognized by Security Analytics. Customized alerts can be created through the Security Analytics reporting function to warn organizations when/if this type of activity occurs.

Figure 10 depicts how Security Analytics is adept at characterizing traffic like potential adversarial lateral movement. The traffic shown depicts multiple SMB sessions between two internal systems. This is just one of the many ways Security Analytics can break traffic down into easily digestible parts allowing traffic to be reviewed from multiple viewpoints.

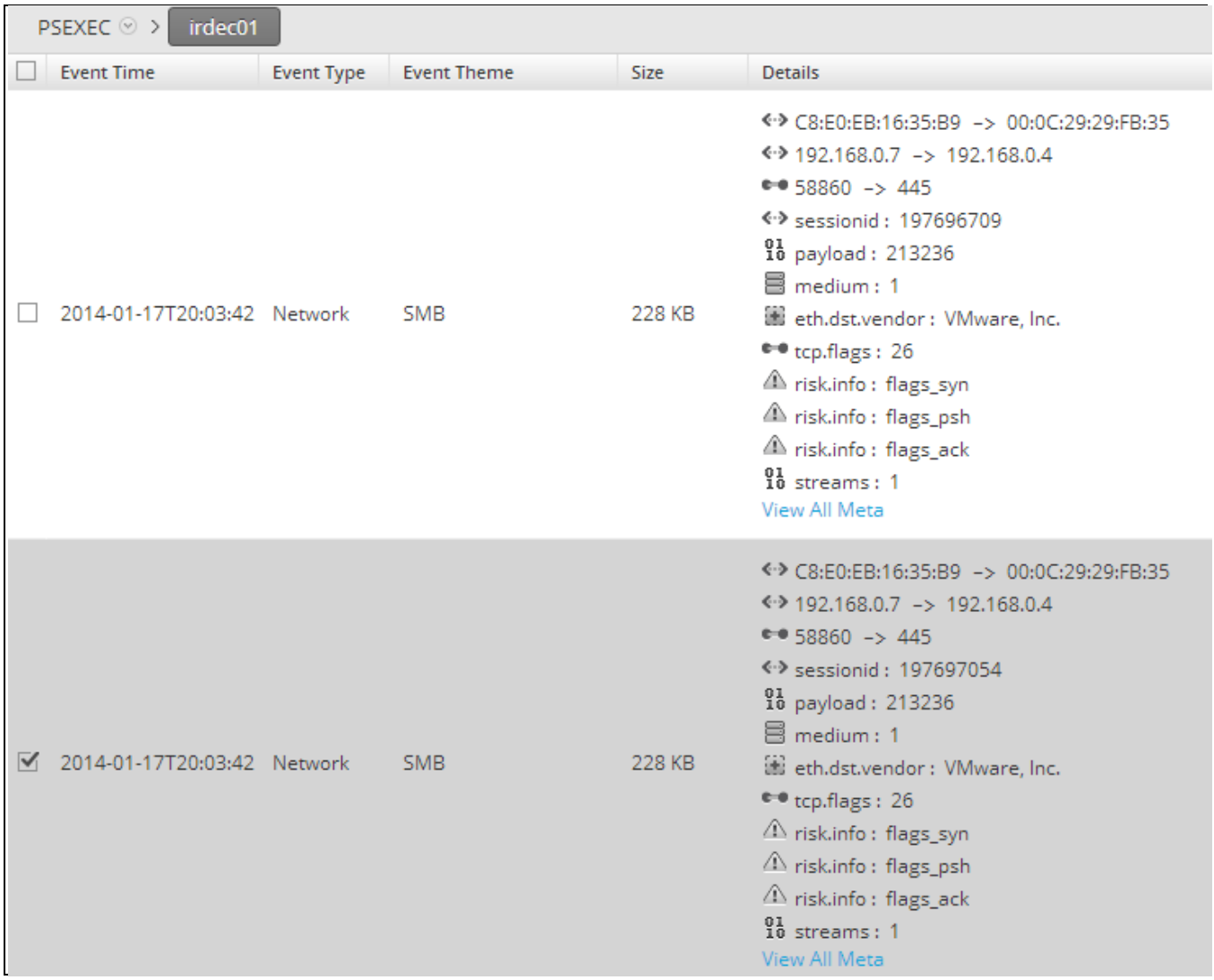

Figure 10: Security Analytics PsExec Detection

Figure 11 further details the PsExec traffic, allowing an organization to quickly triage the traffic to determine if it's malicious. It shows a connection to the Windows ADMIN$ share in which a file called PSEXESVC.exe is transferred. The PSEXESVC.exe is started as a service by PsExec using the Windows Service Control Manager API on the remote system. Once started, PsExec can execute commands on the remote system through the named pipe psexecsvc. This traffic is just one example of how Security Analytics could have alerted an organization during the early stages of an attack.

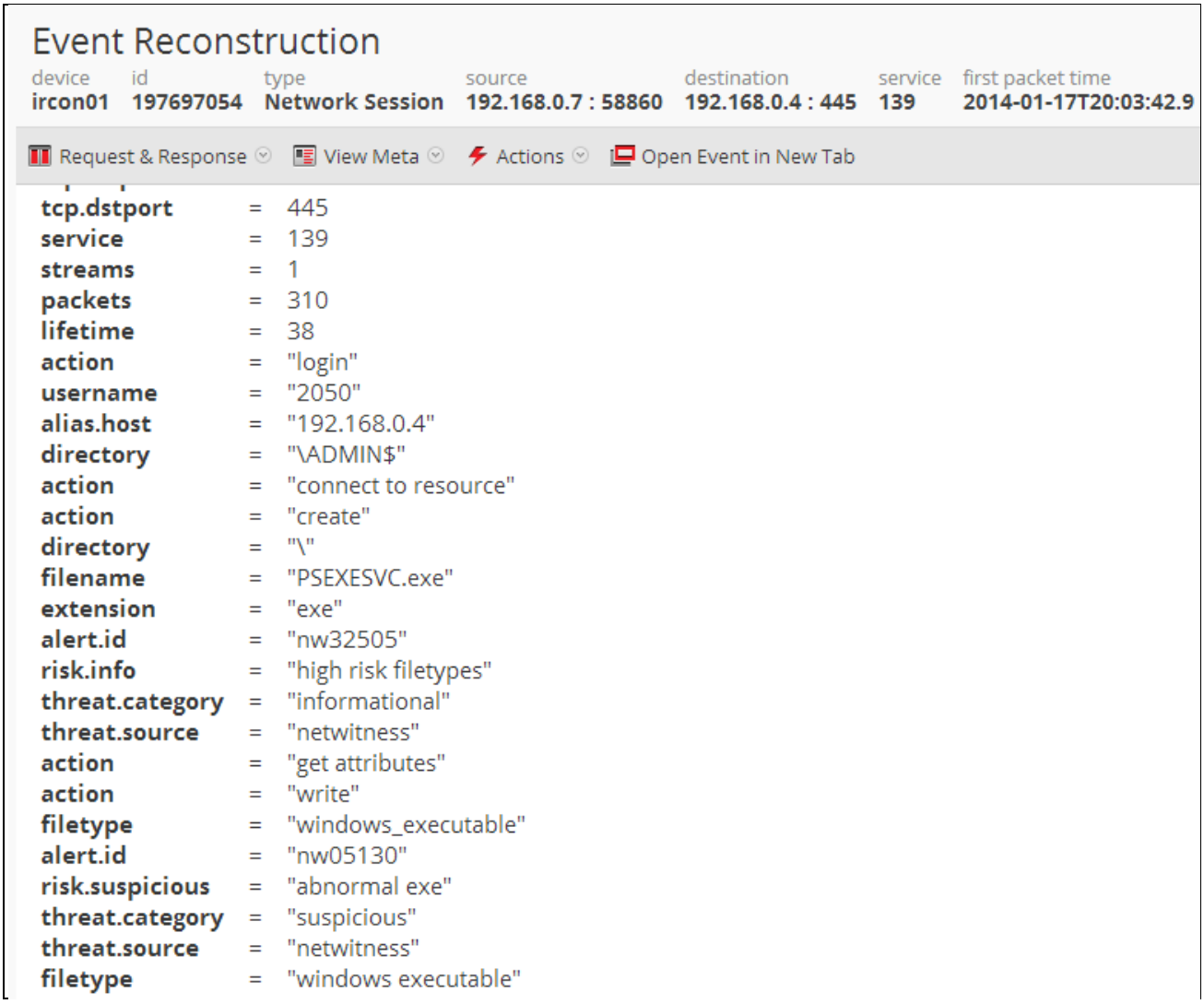

Figure 11: Security Analytics PsExec Detail

3.3 Exfiltration Techniques

In POS attacks and other intrusions, the attackers will often use FTP, as was in the case of the Target breach, as a method of exfiltrating data from a network. Figure 12 is an illustration of how Security Analytics detects and displays this type of traffic. Security Analytics can be customized to alert an organization if this type of traffic is occurring with non-approved IP Addresses.

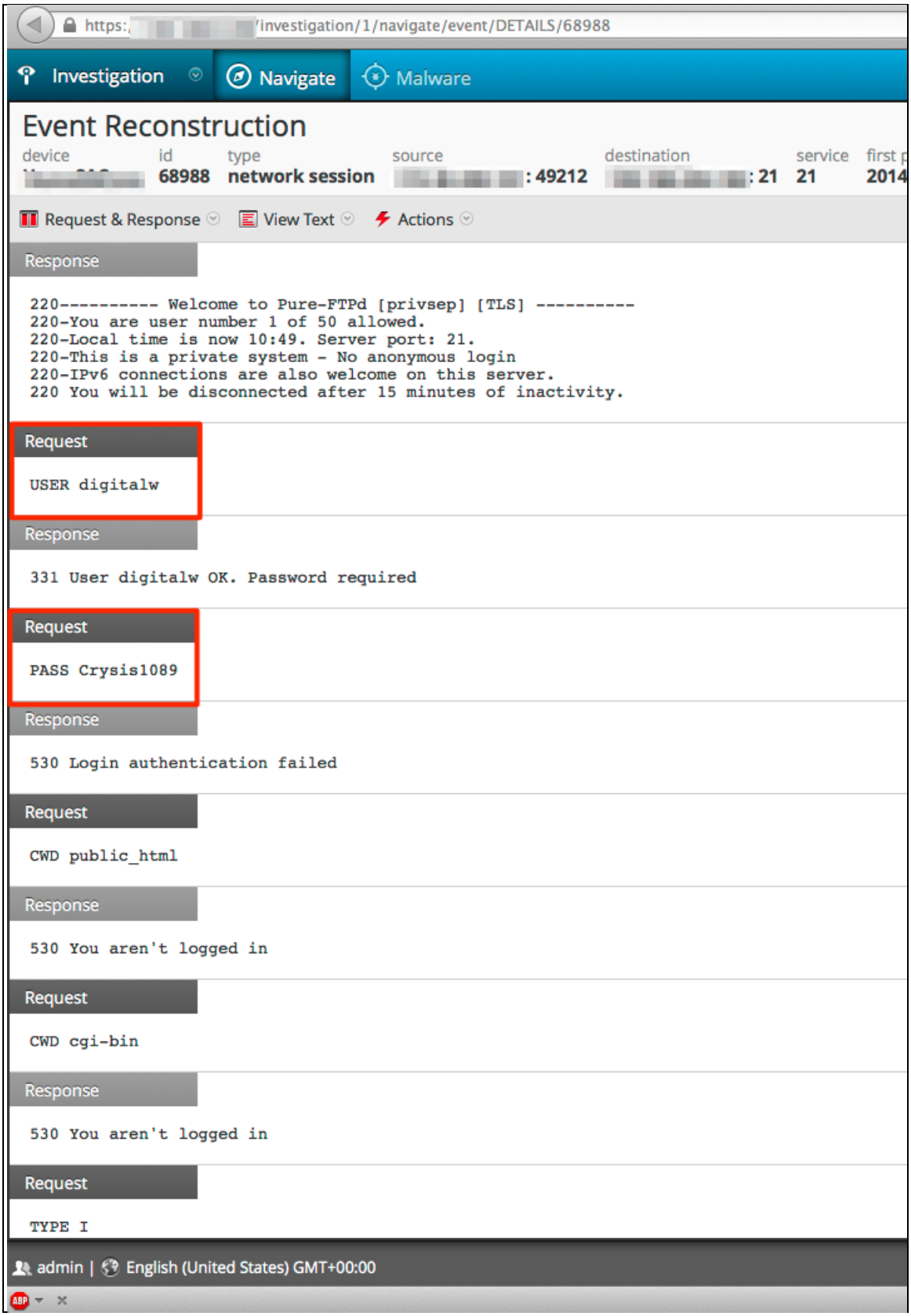

Figure 12: Security Analytics FTP Detection

4. Conclusion

Using tools like to Security Analytics and ECAT in an enterprise can help to identify incidents early, before sensitive data has actually been accessed or exfiltrated. Decreasing the time to detection is critical when dealing with sensitive data, and having this situational awareness can initiate a quicker incident response and reduce exposure. Security Analytics and ECAT provides corporations with the capability to discover and mitigate attacks before they become major incidents. The complements that these tools provide each other not only help analyst locate common problems but assist in sifting through the fog to identify what would have otherwise gone undetected. Focusing on broad and early detection of maliciousness should always be a priority to organizations.


RSA
The Security Division of EMC