Interested in learning
more about security?

SANS Institute
InfoSec Reading Room

Detecting Malicious SMB Activity Using Bro

Attackers utilize the Server Message Block (SMB) protocol to blend in with network activity, often carrying out their objectives undetected. Post-compromise, attackers use file shares to move laterally, looking for sensitive or confidential data to exfiltrate out a network. Traditional methods for detecting such activity call for storing and analyzing large volumes of Windows event logs, or deploying a signature-based intrusion detection solution. For some organizations, processing and storing large amounts of Windows ...

Detecting Malicious SMB Activity Using Bro

GIAC (GCIA) Gold Certification

Author: Richie Cyrus, R_Cyrus@mastersprogram.sans.edu
Advisor: Rob Vandenbrink
Accepted: December 13th 2016

Abstract

Attackers utilize the Server Message Block (SMB) protocol to blend in with network activity, often carrying out their objectives undetected. Post-compromise, attackers use file shares to move laterally, looking for sensitive or confidential data to exfiltrate out a network. Traditional methods for detecting such activity call for storing and analyzing large volumes of Windows event logs, or deploying a signature-based intrusion detection solution. For some organizations, processing and storing large amounts of Windows events may not be feasible. Pattern based intrusion detection solutions can be bypassed by malicious entities, potentially failing to detect malicious activity. Bro Network Security Monitor (Bro) provides an alternative solution allowing for rapid detection through custom scripts and log data. This paper introduces methods to detect malicious SMB activity using Bro.

1. Introduction

"Server Message Block (SMB) can be defined as a protocol used for network file sharing that allows applications on a computer to read and write to file shares and request services from server programs on a computer network." (Microsoft, 2013). Applications can utilize the SMB protocol to retrieve files and resources on a remote server (Microsoft, 2013). SMB primarily uses TCP port 445 for communication, occasionally using TCP port 139 on legacy systems. This paper covers Microsoft SMB version two, supporting Windows 7 and Server 2008 R2 (Walkes/Wireshark, 2016).

Historically, attackers have used SMB to execute commands, search for sensitive or confidential data, and pivot through remote networks. In the early 2000's, attackers used the $IPC share (utilized by a client to send commands to a server), to gain access to systems exposed on the internet (Kan, 2003). More recently, a worm was used in an attack against Sony Pictures, moving throughout their network using SMB shares (Lennon, 2014). Attacks utilizing SMB tend to blend in with file sharing activity on a network, making them difficult to detect.

Identifying such attacks with Windows requires one to enable additional logging via Group Policy Object (GPO) settings and creates a high volume of events to sift through (Microsoft, 2013). Traditional signature-based intrusion detection systems (IDS), such as Snort, can detect malicious SMB activity. However, IDS solutions need to properly normalize packet data to match pattern-based signatures, and can miss attacks using evasion techniques. Bro Network Security Monitor (Bro) provides an alternative solution that allows for rapid detection through custom scripts and log data.

Bro is an open source network security framework based on Unix, and can be used as an intrusion detection system (Bro, 2014). Bro passively inspects traffic on a network, using application protocol analyzers to generate log files with metadata of observed activity (The Bro Project, 2016). Bro log files are in ASCII format, which makes them easy to parse and ingest into a security information and event management (SIEM) solution (The Bro Project, 2016). Bro can detect activity using traditional IDS pattern matching, or event-based detection through the scripting language (The Bro

Richie Cyrus, R_Cyrus@mastersprogram.sans.edu

Project, 2016). When Bro detects malicious activity, it can send an alert to a log or an email address, or spawn an external application in response (The Bro Project, 2016). Bro version 2.5 provides support for the SMB protocol (versions 1 & 2), and is the version referenced in this paper (Johanna/The Bro Project, 2016).

2. Attacker's Use of SMB

The SMB protocol enables files and printers to be shared on a network. Attackers use SMB for a variety of malicious purposes, as they attempt to blend within a victim's network. Once attackers have compromised a system on a network, they can use SMB to connect to file shares and additional systems. They can also use malware that utilizes SMB to spread throughout a network.

2.1. Lateral Movement

Lateral movement can be defined as "techniques that enable an adversary to access, control, and gather information about remote systems on a network" (The MITRE Organization, 2016, Lateral Movement). One of the objectives of lateral movement is for attackers to locate specific files or information of interest to them by pivoting through systems on a compromised network.

2.1.1. SMB Used for Lateral Movement

Threat actors in the past and present have leveraged SMB to carry out

attacks using lateral movement techniques. There are documented examples that show the use of SMB in attacks that were not detected initially. In 2013, threat actors involved in the Ke3chang campaign infected computers on several European Ministries of Foreign Affairs (MFAs) networks and gathered additional information. With that information, they mapped network file shares and copied additional malware to other machines, which enabled them to move laterally (Villeneuve et al., 2014, p. 17). Lazaurus Group used targeted malware to connect to port 139 or 445 used for Windows file sharing. Once connected, they attempted to access the ADMIN$ share on each compromised system. They used the malware to move laterally and spread additional malware to systems (Novetta, 2016).

Richie Cyrus, R_Cyrus@mastersprogram.sans.edu

2.2   Malware Utilizing SMB

Traditionally, malware has often exploited weak SMB configurations. For instance, the Reign malware platform copied and executed malware on computers via Administrative shares. (Kaspersky, 2014). In addition, the Net Crawler malware used in Operation Cleaver, scanned for open SMB ports and utilized an SMB brute force technique to gather cached credentials on compromised networks (Cylance, 2014). Further, the BlackEnergy malware used a plugin named vsnet, which utilized SMB to spread across a network. (Kurt Baumgartner & Maria Garnaeva, 2014). Both Net Crawler and BlackEnergy used the Windows administration tool PsExec to connect to different systems on a network via SMB, using administrative credentials. Most recently, the Locky ransomware variant has been observed using SMB to discover and encrypt network shares on a network (Abrams, 2016). SMB is legitimately used to provide file sharing functionality however; misconfigurations can allow malware to propagate and cause harm to a network.

3. Detecting Malicious SMB Activity

Identifying malicious use of the SMB protocol can be difficult as it blends in with normal file share activity. There are methods to assist with detecting attackers use of SMB: analysis of Windows logs and deploying intrusion detection systems.

3.1 Traditional Methods of Detection

There are two primary solutions used today with the goal of detecting malicious SMB usage in real time: Windows Logging and Snort.

3.1.1 Windows Logging

Corporate networks today primarily use Windows as the operating system for endpoints and servers, which is an advantage as it relates to SMB. SMB logging via group policy objects allows organizations to send SMB-related Window events to a centralized location such as a SIEM solution. A SIEM can make detection easier by analyzing and correlating events to determine if an attack is taking place. Using the Group Policy Management Editor, each system on a domain can generate events when a

Richie Cyrus, R_Cyrus@mastersprogram.sans.edu

file share is accessed, by modifying the default Audit File Share setting (Microsoft, 2013). When events are created using the Audit File Share setting, analysts can more easily detect the use of hidden shares such as $ADMIN and C$, which are commonly seen in attacks.

In addition to the Audit File Share setting, additional logging can be enabled to detect the use of PsExec. PsExec is an administrative tool that executes processes on remote systems with the use of domain credentials. Attackers use PsExec to move laterally on a network by spawning the process cmd.exe, giving them command line access to another system. The Detailed File Auditing GPO setting, when enabled, can detect PsExec's use of the IPC$ share and creation of the PSEXECSVC-* service (Bianco, 2016). Additionally, logging account logins aids in identifying use of the "pass the hash" technique, in which the attacker uses the Windows NLTM hashed password of a user account to access a system. Windows solutions for detection can be useful; however, the employment of them requires a large volume of logging. It can be costly for an organization to store the amount of logging needed for detection. In addition, an organization will need an analyst who has the skillset to parse through the Windows logs to find malicious activity.

3.1.2 SNORT

In the realm of network security monitoring, Snort can be used to detect malicious SMB activity. Snort is an open-source intrusion detection and prevention system, designed to detect attacks via a pattern-matching signature. Below is an example of a Snort rule designed to detect the use of PsExec (Emerging Threats, 2011):

```
alert tcp any any -> $HOME_NET [139,445] (msg:"ET POLICY PsExec? service
created"; flow:to_server, established; content:"|5c 00 50 00 53 00 45 00 58 00 45 00 53
00 56 00 43 00 2e 00 45 00 58 00 45|"; reference:url, xinn.org/Snort-psexec.html;
reference:url, doc.emergingthreats.net/2010781; classtype:suspicious-filename-detect;
sid:201781; rev:2;)
```
Figure 1. Example of a Snort Rule

Some Snort rules created under the General Public License are designed to detect attempts to access and use the C$, $ADMIN, and IPC$ shares on a system. When Snort is deployed on a network, in which the rules mentioned above are enabled, it can provide

Richie Cyrus, R_Cyrus@mastersprogram.sans.edu

some detection for SMB-based attacks. However, this solution can generate a significant number of false positives as system administrators complete tasks, and servers provide file sharing services on a corporate network. The utilization of administrative/hidden shares is not a clear indication of an attack.

3.2 Detection Using Bro

Bro presents an alternative method of detection through network security monitoring. Bro version 2.5 provides new detection capabilities by way of an SMB protocol analyzer (Bro, 2016). The analyzer provides insight into files transferred over SMB, SMB commands, SMB trees, NTLM activity, as well as Distributed Computing Environment (DCE)/ Remote Procedure Call (RPC) activity. Below are the log files introduced in Bro version 2.5 (Bro, 2016):

Log File	Description
dce_rpc.log	Distributed Computing Environment/RPC
ntlm.log	NT LAN Manager
smb_cmd.log	SMB Commands
smb_files.log	SMB Files
smb_mapping.log	SMB Mapping

With Bro, defenders can now build additional detections, and generate alerts on instances of suspicious SMB activity. Bro offers the advantage of detecting activity in real time by passing traffic to the analysis engine. Bro can also detect malicious activity in packet captures, which can be applied retrospectively to past events.

3.3 Testing Environment

The examples, data, and traffic discussed in this paper were generated within the test environment shown in Figure 2 below. VMWare ESXi hosts the following virtual machines, which replicates a corporate environment: Security Onion, Windows Server 2012, Window 7 (Staff machine), Windows 7 (Staff machine). The server running Windows Server 2012 is the domain controller for both Windows 7 systems, which are part of a domain named "UTPROD". The Security Onion distribution is monitoring all ingress and egress traffic from the systems on the UTPROD domain through port mirroring on the switch. The attacker running Kali Linux is assumed to be on the network post-compromise, within the testing environment.

Richie Cyrus, R_Cyrus@mastersprogram.sans.edu
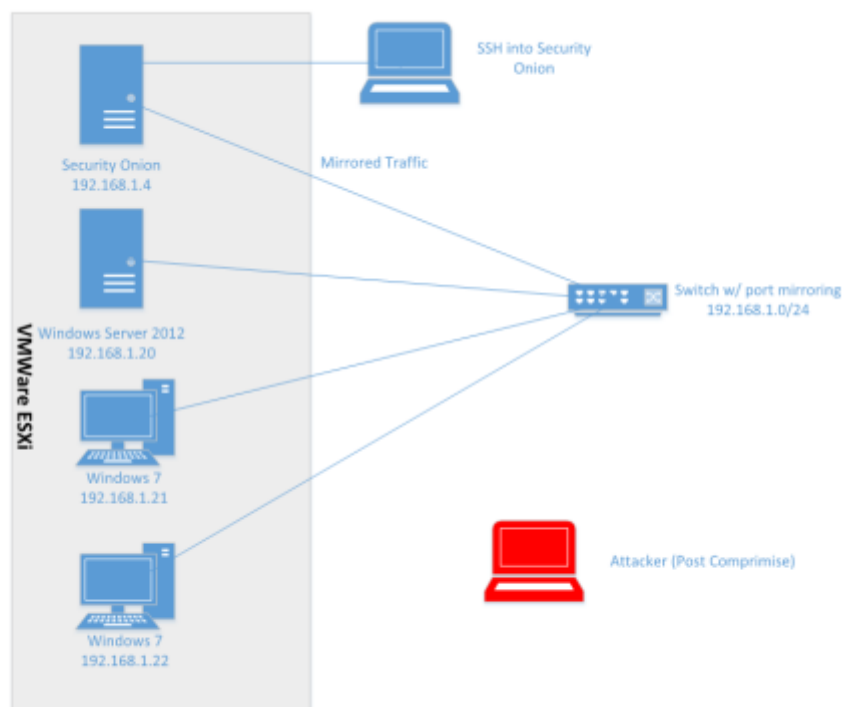
Figure 2. Diagram of lab network

4. Bro Scripting for Detection

The ability to create custom Bro scripts to fit an organization's environment makes Bro a flexible network security monitoring solution. The protocols analyzed by Bro extract metadata used for scripting. The Bro Scripting Framework is based on C++. After reviewing the Bro logs generated by network traffic, script building occurs using string pattern or event-based indicators. Scripts can be configured to generate a "notice" which alerts an analyst to an adverse event on a network.

4.1 Bro Scripts

Bro loads the scripts in the $PREFIX/bro/share/policy/ directory by default, where $PREFIX corresponds the root directory used in the Bro installation. Loading custom scripts requires adding an entry to the local.bro file with the directory path of the script. The local.bro file is located in the $PREFIX/bro/share/policy/site/ directory. Running the command "broctl deploy" from the $PREFIX/bro/bin directory deploys a custom script

Richie Cyrus, R_Cyrus@mastersprogram.sans.edu

making it active. All enabled scripts can be displayed by running the "broctl scripts" command.

Scripts loaded by Bro generate ASCII-based logs, which are populated with network metadata related to connections, files, and protocols. Viewing the various log files can assist analysts in finding malicious traffic on a network, which requires analysis. During analysis, if correlations are made that identify malicious activity, they can be used to create scripts. These scripts, in turn, can notify an analyst to malicious activity as it occurs on a network. Using the Bro Notice framework, alerts may be sent via email or as a log entry in the notice.log file.

4.1.1 – Detection Scripts

The scripts in this section were created to detect potentially malicious SMB activity on a network. Tracking malicious files sent via SMB assists analysts in identifying a potential incident requiring response. The script shown in Figure 3 uses the files analysis (base/frameworks/files) and hash (base/frameworks/hash-all-files) frameworks in Bro to identify files transferred via SMB, and checks their cryptographic hashes against Virus Total's anti-virus database. The script requires an API key from Virus Total, which can be obtained by registering for a free account. If the submitted hash of a file is identified as malicious by two or more anti-virus vendors, a log entry is added to notice.log, alerting an analyst to a potential incident on a network. Searching Virus Total for the hashes of files, rather than submitting the files themselves, does not disclose the sensitive analysis work that is currently in progress. Submitting a file that is part of a targeted attack, may inform attackers that they are discovered on a network, giving them a chance to alter their tactics or cause further damage. Trusted systems should be added to the global trustedIPs variable shown in the script below, which would prevent them from being misidentified as malicious.

```
@load base/frameworks/files
@load base/frameworks/notice
@load frameworks/files/hash-all-files

export {
  redef enum Notice::Type += {
    SMB
  };

global trustedIPs: set[addr] = {192.168.1.22,192.168.1.20} &redef;
```

Richie Cyrus, R_Cyrus@mastersprogram.sans.edu

```
        # url needed to use VirusTotal API
    const vt_url = "https://www.virustotal.com/vtapi/v2/file/report" &redef;

    # VirusTotal API key
    const vt_apikey = "<---- Enter your Virus Total API key here ---->" &redef;

    # threshold of Anti-Virus hits that must be met to trigger an alert
    const notice_threshold = 2 &redef;

event file_hash(f: fa_file, kind: string, hash: string)
{
                # If the file "f" for the event has a source type, and if the source type equals
SMB, check file hash against VirusTotal
                if ( f?$source && f$source == "SMB")
                {
                        local data = fmt("resource=%s", hash);

                        local key = fmt("-d apikey=%s",vt_apikey);

                        # HTTP request out to VirusTotal via API
        local req: ActiveHTTP::Request = ActiveHTTP::Request($url=vt_url,
$method="POST",$client_data=data, $addl_curl_args=key);

        when (local res = ActiveHTTP::request(req))
                {
                        if ( |res| > 0 )
                        {
                                if ( res?$body )
                                {
                                        local body = res$body;

                                local tmp = split_string(res$body,/\}\},/);

                                        if ( |tmp| != 0 )
                                        {
                                                local stuff = split_string( tmp[1], /\,/ );


                                                # splitting the string that contains the amount of
positive anti-virus hits on ":" "positives:23"

                                                local pos = split_string(stuff[9],/\:/);

                                                # converting the string from variable pos into a
integer

                                                local notic = to_int(pos[1]);

                                                # If the number of positives (number stored in
variable notic) equals or exceeds the threshold, generate a notice
                                                if (notic >= notice_threshold )
                                                {
```

Richie Cyrus, R_Cyrus@mastersprogram.sans.edu

```
                                                      local msg = fmt("%s,%s,%s","Potentially
Malicious File Transfered via SMB",stuff[9],stuff[4]);

                                                      local n: Notice::Info = Notice::Info($note=SMB,
$msg=msg, $sub=stuff[5]);


            Notice::populate_file_info2(Notice::create_file_info(f), n);
                                            if (n$id$orig_h !in trustedIPs){
                                                    NOTICE(n);
                                                    }
                                                    }
                                        }
                    }

                    }

            }
        }
    }
}
```

Figure 3. Bro Script Detecting the Use of Malicious Files in SMB Traffic

Figure 4 displays another custom script, created to detect the use of the C$, ADMIN$, or
IPC$ shares. While there are legitimate uses for these shares, activity involving these
shares should be limited. Attackers use these shares to execute services and processes,
upload/transport malware, and move laterally. Any systems seen in the notice log for this
alert should be investigated to determine if they are infected or compromised. Trusted
administrative systems can be tuned out by adding their static IP to the trustedIPs set
below.

```
@load base/frameworks/files
@load base/frameworks/notice
@load policy/protocols/smb

export {
   redef enum Notice::Type += {
      Match
   };

global isTrusted = T;

function hostAdminCheck(sourceip: addr): bool
```

Richie Cyrus, R_Cyrus@mastersprogram.sans.edu

```
{
        if (sourceip !in trustedIPs)
        {
                return F;
        }
        else
        {
                return T;
        }

}


event smb2_tree_connect_request(c: connection, hdr: SMB2::Header, path: string)
{

isTrusted = hostAdminCheck(c$id$orig_h);
if (isTrusted == F){

        if ("IPC$" in path || "ADMIN$" in path || "C$" in path)
        {
                NOTICE([$note=Match, $msg=fmt("Potentially Malicious Use of an
Administative Share"), $sub=fmt("%s",path), $conn=c]);
        }
}
}

event smb1_tree_connect_andx_request(c: connection, hdr: SMB1::Header, path: string, service:
string)
{
isTrusted = hostAdminCheck(c$id$orig_h);
if (isTrusted ==F){

if ("IPC$" in path || "ADMIN$" in path || "C$" in path)
{
NOTICE([$note=Match, $msg=fmt("Potentially Malicious Use of an Administative Share"),
$sub=fmt("%s",path), $conn=c]);
}
}
}

}
```

Figure 4. Bro Script Detecting the Use of Hidden SMB Shares

The custom script in Figure 5 detects usage of NTLM traffic using a host name that does

not match an organization's standard naming convention. If a network's computers are on

Richie Cyrus, R_Cyrus@mastersprogram.sans.edu

a domain and have a standard naming convention, any activity seen on the network with a host name outside the convention is potentially malicious. The activity may indicate that an external entity is using domain credentials to navigate to various systems within the network. The comparison variable in this script contains the string "WIN7PROD", representing the part of the naming convention used throughout the network:

```
@load base/frameworks/notice
@load policy/protocols/smb

export {
  redef enum Notice::Type += {
    SMB
  };

event ntlm_authenticate(c: connection, request: NTLM::Authenticate)
{
# strip out the first 5 characters of workstation value to be compared to naming
convention
local strcheck = sub_bytes(request$workstation, 1, 8);

# value of the comparison of the two strings
local comp_str = strcmp(strcheck, "WIN7PROD");

    # If the comparison of the strings stored in variable comp_str are not the same,
generate a notice.
    If (comp_str != 0 )
    {
    NOTICE([$note=SMB, $msg=fmt("Potential Lateral Movement Activity – Invalid
Hostname using Domain Credentials"), $sub=fmt("%s,%s","Suspicious Hostname:",
request$workstation), $conn=c]);
    }
}
}
```
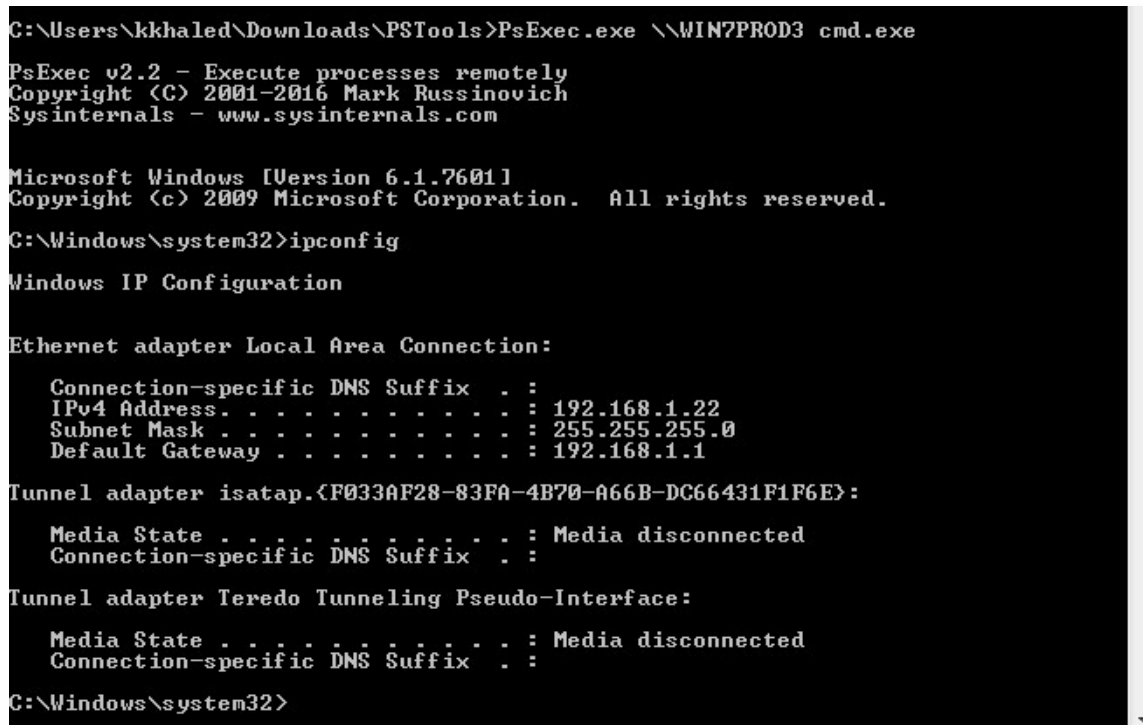
Figure 5. Bro Script Detecting the Use of A Rogue Hostname In SMB Traffic

4.2. Examples of Detection

Below are some examples of the capabilities that the scripts shown in Figures 3, 4, and 5 provide to detect attacks utilizing SMB.

4.2.1 – PsExec

Richie Cyrus, R_Cyrus@mastersprogram.sans.edu

With administrative credentials, attackers can use PsExec to execute processes on a remote computer. The host in Figure 6 is compromised. PsExec was used by the attacker to gain command line access to another computer on the lab network (WIN7PROD3) by executing cmd.exe.


```
C:\Users\kkhaled\Downloads\PSTools>PsExec.exe \\WIN7PROD3 cmd.exe

PsExec v2.2 - Execute processes remotely
Copyright (C) 2001-2016 Mark Russinovich
Sysinternals - www.sysinternals.com


Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.1.22
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1

Tunnel adapter isatap.{F033AF28-83FA-4B70-A66B-DC66431F1F6E}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter Teredo Tunneling Pseudo-Interface:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Windows\system32>
```

Figure 6. PsExec Usage to Connect to System WINPROD3

Using the custom Bro scripts, PsExec's use of the ADMIN$ and IPC$ shares was detected, which added notice messages of "Potentially Malicious Use of an Administrative Share". The use of PsExec creates an executable named PSEXESVC.exe. PSEXESVC.exe was identified as potentially malicious using Virus Total data generating the notice message "Potentially Malicious File Transferred via SMB", as seen in Figure 7 below. The notice message includes the number of anti-virus vendors that classify the file as malicious, as well as a URL to the file's Virus Total analysis results.

Richie Cyrus, R_Cyrus@mastersprogram.sans.edu

Figure 7. Bro Notice.log after PsExec Activity

4.2.2 – PsExec Alternatives

WmiExec uses Windows Management Instrumentation (WMI) to execute code and commands on a remote system. (Kennedy, 2015). In Figure 8, WmiExec is used by the attacker to launch a semi-interactive shell on the remote lab system at 192.168.1.22. Domain (UTPROD) administrator credentials were used to gain access.

Figure 8. Wmiexec Usage

Bro detects WmiExec's use of the ADMIN$ file share shown in Figure 9. A message was added to the notice log outlining the systems involved and the file share seen, which in

Richie Cyrus, R_Cyrus@mastersprogram.sans.edu

this case is \\\\192.168.1.22\\ADMIN$. The source IP in the notice log message is 192.168.1.6 belonging to the attacker using WmiExec.

Figure 9. Bro Notice.log following WmiExec Activity

Figure 10 shows the attacker using SmbExec to spawn a semi-interactive command shell on the system using IP address 192.168.1.21.

Figure 10. SmbExec usage

Bro detects SmbExec's default use of the C$ share to spawn a command shell on the remote system, by using the script shown Figure 4. A notice message was generated which shows the source IP of the attacker (192.168.1.6), the victim IP address (192.168.1.21), and the file share accessed by the attacker \\\\192.168.1.21\\C$.

Richie Cyrus, R_Cyrus@mastersprogram.sans.edu

Figure 11. Bro Logs following the SmbExec Activity

4.2.3 – Metasploit psexec

Metasploit is a popular framework used for penetration testing, and contains a modified version of PsExec.exe. Figure 12 shows the attacker's configured options for the PsExec Metasploit module within Metasploit's msfconsole.

Richie Cyrus, R_Cyrus@mastersprogram.sans.edu

Figure 12. Metasploit PsExec Module to Connect to 192.168.1.22

The "exploit" command runs the module which results in a meterpreter session, giving the attacker access to the remote system at IP address 192.168.1.22 as seen in Figure 12. Figure 13 displays the Bro notice log, verifying detection of Metasploit PsExec module use of ADMIN$ and IPC$ shares, as well as use of the random hostname "Nhyl80UC9iXFEcJH".

Figure 13. Bro Logs following The Metasploit PsExec Activity

Attackers are known to transfer or upload additional malware to file shares during attacks. Figure 14 shows the tool Mimikatz (renamed to "badfile"), which is used to dump passwords from memory, being uploaded to a remote system in the lab environment via the C$ share.

Richie Cyrus, R_Cyrus@mastersprogram.sans.edu
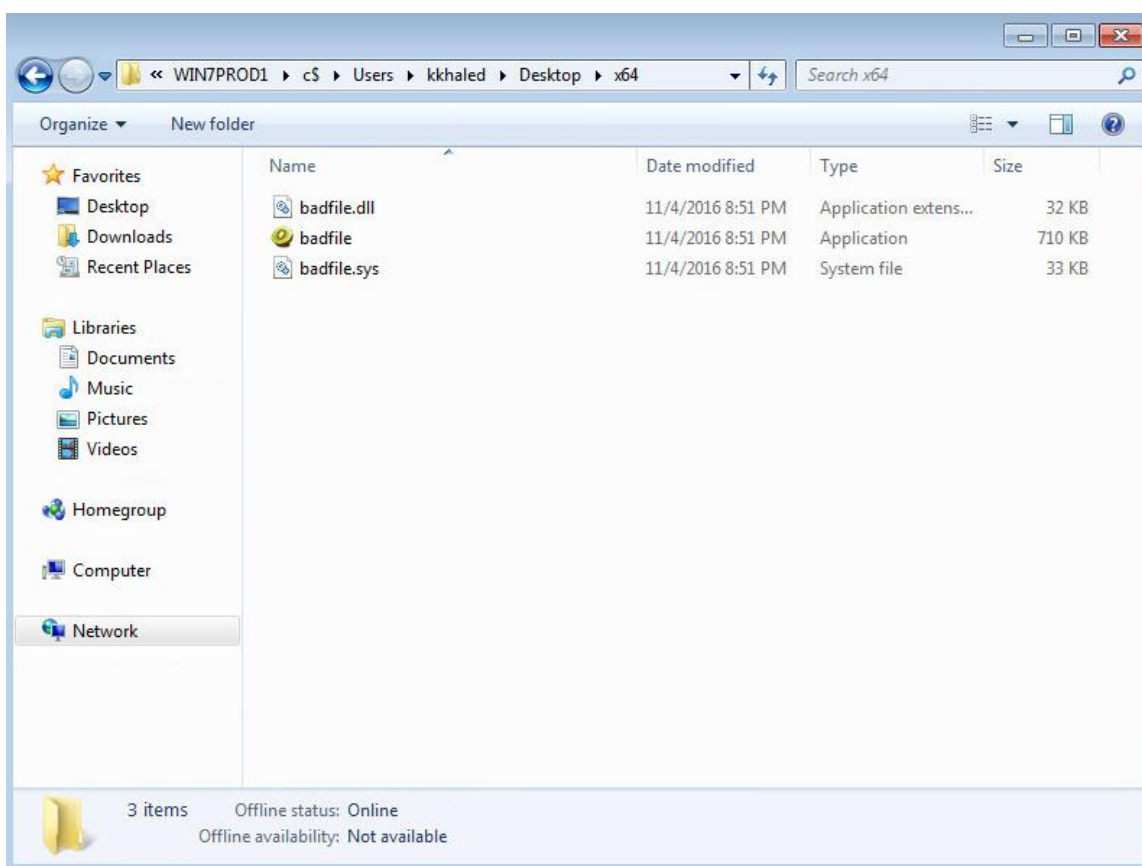
Figure 14. Mimikatz Uploaded to the C$ on a Remote System

Bro detects that Mimikatz was transferred over SMB and checks its hash against Virus Total. Mimikatz is identified by thirty-five anti-virus vendors as being malicious, thus generating the alerts seen in Figure 15. The text displayed in the notice log "positives:35" relates to the number of anti-virus vendors that categorize Mimikatz as malicious.

Figure 15. Bro Notice.log following the Upload of Mimikatz to the C$ share

Richie Cyrus, R_Cyrus@mastersprogram.sans.edu

5.   Conclusion

Attackers use the SMB protocol in ways that blend in with day-to-day network traffic. These malicious entities then move laterally within a network, post-compromise, and attempt to access systems looking for sensitive data. The SMB protocol allows their activity hard to detect. Collecting Windows event logs related to file share auditing is a method for detecting malicious SMB activity, however this is not ideal due to the large volume of logs generated. Intrusion detection systems, such as Snort rely primarily on pattern-based indicators, which can be bypassed and may be difficult to tune. Bro Network Security Monitor can analyze the SMB protocol and provide metadata which can be used to identify potential indicators of compromise. These indicators are the basis of scripts that are used to detect malicious activity and alert analysts. The scripts introduced in this paper generate alerts when potentially malicious files transferred via SMB, hidden file shares such as C$ are used, and when suspicious hostnames seen in SMB traffic. Bro proves to be an effective, open-sourced, and cost efficient, solution to detect and respond to malicious activity using SMB.

Richie Cyrus, R_Cyrus@mastersprogram.sans.edu

6. References

References

Abrams, L. (2016). The Locky Ransomware Encrypts Local Files and Unmapped

Network Shares. Retrieved from Bleeping Computer website:

http://www.bleepingcomputer.com/news/security/the-locky-ransomware-

encrypts-local-files-and-unmapped-network-shares/

Bianco, D. J. (2016, August 3). ThreatHunting/psexec-windows-events.md at master ·

ThreatHuntingProject/ThreatHunting · GitHub. Retrieved October 29, 2016,

from

https://github.com/ThreatHuntingProject/ThreatHunting/blob/master/hunts/psexe

c-windows-events.md

The Bro Project. (2016, November 11). Introduction — Bro 2.4.1 documentation.

Retrieved November 12, 2016, from https://www.bro.org/sphinx/intro/index.html

The Bro Project. (2016, August 17). Release Notes — Bro 2.5-beta-114 documentation.

Retrieved October 30, 2016, from https://www.bro.org/sphinx-git/install/release-

notes.html#new-dependencies

Bro. (2014). The Bro Network Security Monitor. Retrieved from https://www.bro.org/

Cylance. (2014). Cylance Operation Cleaver Report. Retrieved from

https://cdn2.hubspot.net/hubfs/270968/assets/Cleaver/Cylance_Operation_Cleav

er_Report.pdf

Richie Cyrus, R_Cyrus@mastersprogram.sans.edu

Emerging Threats. (2011, October 12). 2010781 < Main < EmergingThreats. Retrieved

    October 29, 2016, from http://doc.emergingthreats.net/bin/view/Main/2010781

Johanna/The Bro Project. (2016, August 18). Bro Blog: Bro 2.5 Beta. Retrieved

    November 12, 2016, from http://blog.bro.org/2016/08/bro-25-beta.html

Kan, B. (2003). IPC Share Exploit: Methodology of Chinese Attackers. Retrieved from

    SANS Insitute website: https://www.giac.org/paper/gcih/466/ipc-share-exploit-

    methodology-chinese-attackers/103860

Kaspersky. (2014). THE REGIN PLATFORM NATION-STATE OWNAGE OF GSM

    NETWORKS. Retrieved from

    https://securelist.com/files/2014/11/Kaspersky_Lab_whitepaper_Regin_platform

    _eng.pdf

Kennedy, D. (2015, June 12). We Don't Need No Stinkin' PSExec - TrustedSec -

    Information Security. Retrieved from https://www.trustedsec.com/june-

    2015/no_psexec_needed/

Kurt Baumgartner, & Maria Garnaeva. (2014). BE2 Custom Plugins, Router Abuse, and

    Target Profiles - Securelist. Retrieved from Securelist website:

    https://securelist.com/blog/research/67353/be2-custom-plugins-router-abuse-and-

    target-profiles/

Lennon, M. (2014, December 19). Hackers Used Sophisticated SMB Worm Tool to

    Attack Sony | SecurityWeek.Com. Retrieved from

Richie Cyrus, R_Cyrus@mastersprogram.sans.edu

http://www.securityweek.com/hackers-used-sophisticated-smb-worm-tool-attack-sony

Microsoft. (2013, July 3). Audit File Share. Retrieved October 29, 2016, from https://technet.microsoft.com/en-us/library/dn311489(v=ws.11).aspx

Microsoft. (2013, June 24). Server Message Block Overview. Retrieved November 12, 2016, from https://technet.microsoft.com/en-us/library/hh831795.aspx

The MITRE Organization. (2016). Lateral Movement - ATT&CK. Retrieved from https://attack.mitre.org/wiki/Lateral_Movement

Novetta. (2016). Operation Blockbuster RAT and Staging Report. Retrieved from https://www.operationblockbuster.com/wp-content/uploads/2016/02/Operation-Blockbuster-RAT-and-Staging-Report.pdf

Novetta. (2016). Operation Blockbuster Report. Retrieved from https://www.operationblockbuster.com/wp-content/uploads/2016/02/Operation-Blockbuster-Report.pdf

Villeneuve, N., Bennett, J., Moran, N., Haq, T., Scott, M., & Geers, K. (2014). Operation "Ke3chang". Retrieved from FireEye website: https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/wp-operation-ke3chang.pdf

Walkes/Wireshark, D. (2016, October 23). SMB2 - The Wireshark Wiki. Retrieved November 12, 2016, from https://wiki.wireshark.org/SMB2

Richie Cyrus, R_Cyrus@mastersprogram.sans.edu

Upcoming SANS Training
Click Here for a full list of all Upcoming SANS Events by Location

SANS Zurich 2018	**Zurich, CH**	**Apr 16, 2018 - Apr 21, 2018**	**Live Event**
SANS London April 2018	**London, GB**	**Apr 16, 2018 - Apr 21, 2018**	**Live Event**
SANS Baltimore Spring 2018	**Baltimore, MDUS**	**Apr 21, 2018 - Apr 28, 2018**	**Live Event**
Blue Team Summit & Training 2018	**Louisville, KYUS**	**Apr 23, 2018 - Apr 30, 2018**	**Live Event**
SANS Seattle Spring 2018	**Seattle, WAUS**	**Apr 23, 2018 - Apr 28, 2018**	**Live Event**
SANS Doha 2018	**Doha, QA**	**Apr 28, 2018 - May 03, 2018**	**Live Event**
SANS Riyadh April 2018	**Riyadh, SA**	**Apr 28, 2018 - May 03, 2018**	**Live Event**
SANS SEC460: Enterprise Threat Beta Two	**Crystal City, VAUS**	**Apr 30, 2018 - May 05, 2018**	**Live Event**
Automotive Cybersecurity Summit & Training 2018	**Chicago, ILUS**	**May 01, 2018 - May 08, 2018**	**Live Event**
SANS SEC504 in Thai 2018	**Bangkok, TH**	**May 07, 2018 - May 12, 2018**	**Live Event**
SANS Security West 2018	**San Diego, CAUS**	**May 11, 2018 - May 18, 2018**	**Live Event**
SANS Melbourne 2018	**Melbourne, AU**	**May 14, 2018 - May 26, 2018**	**Live Event**
SANS Northern VA Reston Spring 2018	**Reston, VAUS**	**May 20, 2018 - May 25, 2018**	**Live Event**
SANS Amsterdam May 2018	**Amsterdam, NL**	**May 28, 2018 - Jun 02, 2018**	**Live Event**
SANS Atlanta 2018	**Atlanta, GAUS**	**May 29, 2018 - Jun 03, 2018**	**Live Event**
SANS Rocky Mountain 2018	**Denver, COUS**	**Jun 04, 2018 - Jun 09, 2018**	**Live Event**
SANS London June 2018	**London, GB**	**Jun 04, 2018 - Jun 12, 2018**	**Live Event**
DFIR Summit & Training 2018	**Austin, TXUS**	**Jun 07, 2018 - Jun 14, 2018**	**Live Event**
SANS Milan June 2018	**Milan, IT**	**Jun 11, 2018 - Jun 16, 2018**	**Live Event**
SANS ICS Europe Summit and Training 2018	**Munich, DE**	**Jun 18, 2018 - Jun 23, 2018**	**Live Event**
SANS Cyber Defence Japan 2018	**Tokyo, JP**	**Jun 18, 2018 - Jun 30, 2018**	**Live Event**
SANS Philippines 2018	**Manila, PH**	**Jun 18, 2018 - Jun 23, 2018**	**Live Event**
SANS Oslo June 2018	**Oslo, NO**	**Jun 18, 2018 - Jun 23, 2018**	**Live Event**
SANS Crystal City 2018	**Arlington, VAUS**	**Jun 18, 2018 - Jun 23, 2018**	**Live Event**
SANS Minneapolis 2018	**Minneapolis, MNUS**	**Jun 25, 2018 - Jun 30, 2018**	**Live Event**
SANS Cyber Defence Canberra 2018	**Canberra, AU**	**Jun 25, 2018 - Jul 07, 2018**	**Live Event**
SANS Paris June 2018	**Paris, FR**	**Jun 25, 2018 - Jun 30, 2018**	**Live Event**
SANS Vancouver 2018	**Vancouver, BCCA**	**Jun 25, 2018 - Jun 30, 2018**	**Live Event**
SANS London July 2018	**London, GB**	**Jul 02, 2018 - Jul 07, 2018**	**Live Event**
Pre-RSA® Conference Training	**OnlineCAUS**	**Apr 11, 2018 - Apr 16, 2018**	**Live Event**
SANS OnDemand	**Books & MP3s OnlyUS**	**Anytime**	**Self Paced**