Reliably Detecting Pass the Hash Through Event Log Analysis

Written by David Kennedy

October 12, 2016

At BDS we have the unique ability to pull large subsets of data in order to identify abnormal patterns in environments. With our BDS Vision product, the endpoint is one of the easiest ways for us to identify compromises within an organization and we continue to add better detection capabilities every day. One of the many features we have within Vision is the ability to detect Pass the Hash (PtH) through multiple methods. I recently presented one of the methods we use for Pass the Hash detection at this year's GrrCon. The point of this detection is not to focus on a tool, but rather the behavior of a specific indicator within a network and patterns that are abnormal.

Pass the Hash is still an extremely problematic issue for most organizations and still something that we use regularly on our pentests and red teams. When looking at detecting Pass the Hash, I first started by doing research to see if anyone else has already been reliably detecting pass the hash across the network. There are some excellent articles out there, however nothing that was reliable or without generating a large number of false positives.

I won't go into the history of Pass the Hash or its inner-workings in this blog, but if you are interested, the following is an excellent article SANS Pass the Hash Attacks Mitigation.

Briefly, an attacker needs to grab hashes off from a system, which is usually done through targeted attacks such as spear phishing or directly compromising a host through other methods (For example: TrustedSec Blog on Responder). Once access to a remote system is obtained, the attacker will elevate to SYSTEM level permissions and from there attempt to extract the hashes through multiple methods (registry, process injection, volume shadow copies, etc.). For Pass the Hash, the attacker is typically targeting the LM/NTLM hashes on the system (more commonly NTLM). We can't Pass the Hash using things such as NetNTLMv2 (through responder or other methods) or cached credentials. We need pure and unfiltered NTLM hashes. There are essentially only two locations where we can obtain these credentials; the first is through a local account (such as an administrator RID 500 account, or other local accounts) and the second is domain controllers.

The main exposure with Pass the Hash comes from the fact that most organizations have a shared local account on one system, so we can extract the hashes off that one system and

Recent Posts

- Enhanced Endpoint Protection: FedEx Invoice Variation
- B2B USA Business Spam List
- The Vision Platform Adds Support for OS X and Linux
- Petya Ransomware Without The Fluff
- Binary Defense and the Brakeing Down Security Podcast!

Posts by Topic

- Security (12)
- Vision (9)
- Threat Intel (5)
- Menu Article (4)
- MSSP (2)

see all

move to other systems on the network. There are ways of mitigating this, however they are not 100% reliable. For example, the Microsoft patches and newer versions of Windows (8.1 and Server 2012 R2) offer Pass the Hash, however this only applies to "other" accounts and not the

You can prohibit Pass the Hash through the GPO:

"Deny access to this computer from the network"

Located under:

Computer Configuration\WindowsSettings\Security Settings\Local Policies\User Rights Assignment

Most organizations don't have the ability to implement that GPO, and Pass the Hash possibilities are rampant.

The next question is; how do you detect Pass the Hash?

Pass the Hash is challenging because it exhibits normal behavior within the network. A good example is: what happens when you close an RDP session and your session isn't closed? When you go to re-auth, your machine is still there. That exhibits very similar behavior as Pass the Hash in the network.

By doing extensive testing on log analysis deployed to hundreds of thousands of systems, we've been able to identify very specific behavior that has a low false positive rate within most organizations. There are a number of rules you can add to the following detection capability, for example seeing a number of successes across the network would indicate a spray of Pass the Hash, or multiple failure attempts would indicate failed credentials.

For the following, we want to see all logon type 3 (network logons) with event log 4624. We are looking for the NtLmSsP account (this can be triggered by multiple things) as well as the key length to be set to 0. These indicate lower level protocols that are typically used through Pass the Hash (WMI, SMB, etc.). In addition, since the only two locations we can GET access to hashes are through local hashes or through domain controllers, we can detect Pass the Hash across the network through local accounts by filtering for only local accounts. This means if your domain is GOAT, you would filter anything with GOAT\ in it and alert on the rest. This should eliminate things such as scanners, PSEXEC used by administrators, etc.

Note that you can (and probably should) keep domain in there, but you will most likely need to tune to normal behavior within your infrastructure. A good example is that OWA will have a key length of 0 and have the exact same indicators as Pass the Hash based on its proxy auth. This is normal behavior for OWA and obviously not Pass the Hash. This would not flag if you were filtering on local accounts only.

EventID: 4624
Logon Type: 3
Logon Process: NtLmSsP
Security ID: Null SID - Optional and not required however, haven't seen where Null SID isn't used in PtH.
Hostname: (note that this isn't 100% effective; for example, Metasploit and other similar tools will randomize hostnames). You could potentially import a list of all PCs and if none flag, this would reduce false positives. Note however that this is not a reliable method to cut down false positives. Not all tools do this, and have limited detection capabilities on hostname.
Account Name and Domain Name: Alert on only account names that are only local accounts (i.e. do not include domain\username). This will reduce false positives in the network, however if you alert on all of these, it will detect things such as scanners, psexec, and others, but will take time to tune these out. It's not necessarily a bad thing to flag on all accounts (skip the COMPUTER$ accounts) and tune in your environment for known patterns and investigate unknown ones.

and investigate unknown ones.

Key Length: 0 - This is the session key length. This is one of the most important for detection within event logs. With something like RDP, this would be 128-bit value. Any lower will be 0 which is a better indicator of lower level protocols with no session

An added benefit is that this event log contains the source IP address for the authentication, so you can quickly identify the source of Pass the Hash within the network.

In order to detect this, we first need to ensure we have the appropriate group policy setup. We need to set account logons to "Success" as we need event log 4624 as the method for detection.

Let's break down the logs and the Pass the Hash attack. In this scenario, let's first imagine that the attacker obtains credentials from a compromised box through phishing and elevates themselves to administrative level rights. Getting the hashes off the system is trivial. Let's say the built in administrator account is shared across multiple systems and the attacker wants to move from SystemA (already compromised) to SystemB (not yet compromised but shared administrator account) through Pass the Hash.
We'll use Metasploit psexec in this example, although there are plenty of other methods/tools that accomplish this same goal:

In this example the attacker establishes a connection to the second system through Pass the Hash. Next, let's look at the event log 4624 and what it contains:

Security ID: NULL SID can be an indication, but do not rely off of this as not all tools may use it. I haven't personally seen PtH not use NULL SID, but it may be possible.

Next, the workstation name definitely looks suspicious; this isn't a good indication however, as not all methods do this. You can use this as an added bonus for investigation purposes, but we do not suggest using the workstation name as an indicator. The source network address can be used to track which IP was performing the Pass the Hash technique for investigation purposes.

Next we see the logon process as NtLmSsp and the key length of 0. These are important for Pass the Hash techniques.

Next we see Logon Type 3 (remote login via the network).

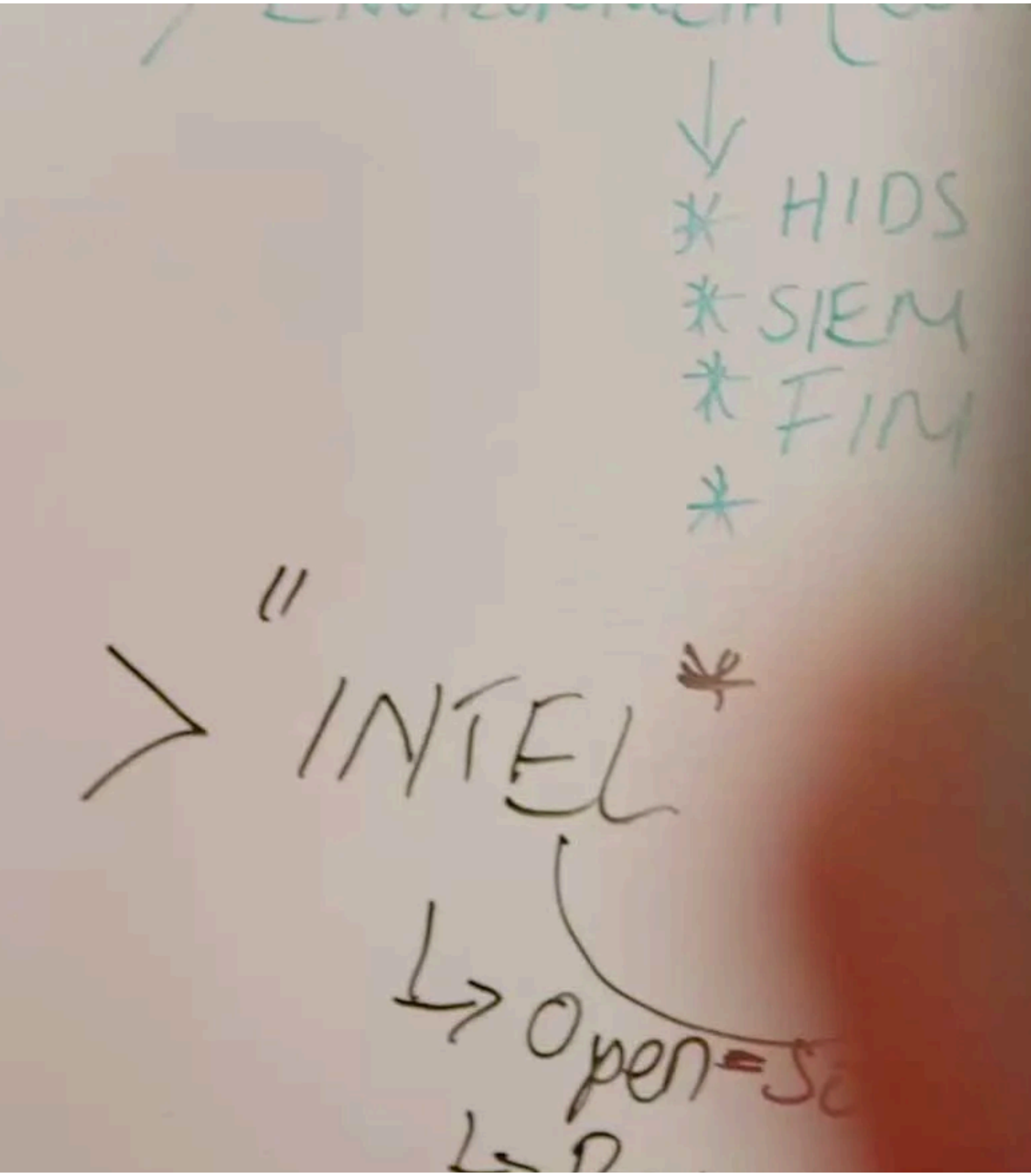
Lastly, we see that this is a local account based on the account domain and the name.

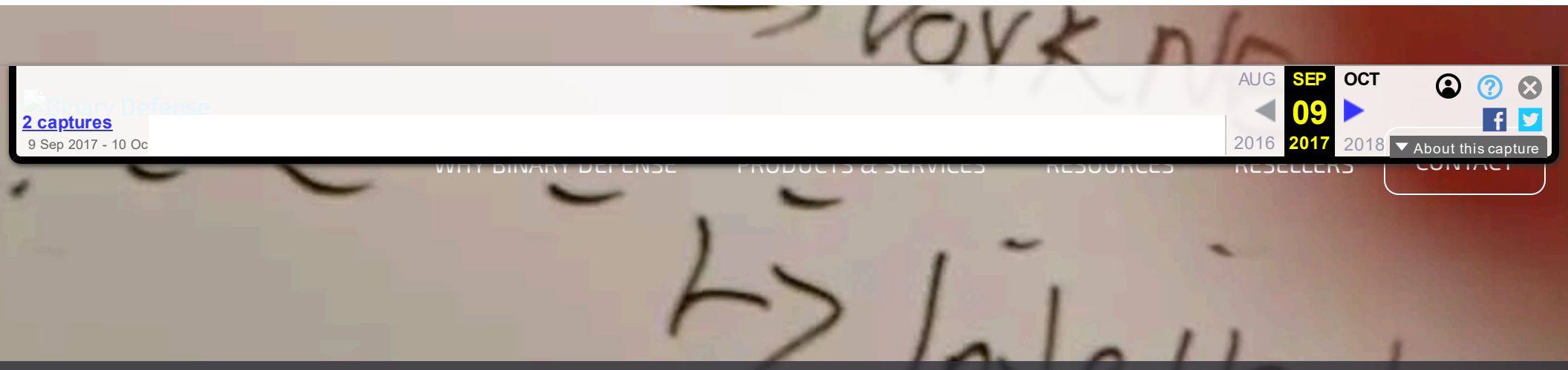
In conclusion, there are a number of ways to detect Pass the Hash within an environment. This one has been effective in both small and large networks and has been extremely reliable based on different use cases using Pass the Hash. It may need tweaking based on your network environment, however it's pretty simple to reduce false positives and investigate during an attack.

Pass the Hash is still widely used across networks and a common problem for most organizations. There are ways to prohibit and reduce the effectiveness of Pass the Hash, however not all organizations can implement this effectively. The next best option is detection.

This article was written by David Kennedy – Founder of Binary Defense Systems and TrustedSec.

Written by David Kennedy





 Privacy Policy 1-800-BINARY2 info@binarydefense.com 5 Aurora St. Hudson, OH 44236-2902 Web Design By Insivia