Global Information Assurance Certification Paper

Beyond the Cookie:
Using Network Traffic Characteristics to Enhance Confidence in User Identity

GIAC Certified Intrusion Analyst (GCIA) Gold Certification

Author: Courtney Imbert, courtneyimbert@gmail.com
SANS Technology Institute MSISE Candidate

Advisor: Antonios Atlasis

Accepted: August 14, 2014

Abstract

Attackers don't always use sophisticated techniques to infiltrate organizations. They may simply purchase or steal passwords and test credentials against points of entry. Multi-factor authentication decreases the risk of an attacker logging in successfully, but doesn't address a critical issue: how can an organization detect an attacker who has already entered the network with legitimate credentials? This paper will explore the concept of combining the attributes of passively intercepted network packets with a user profile to increase the Level of Assurance (LOA) in a user's unique identity. A user profile developed from network behavior can be used to detect an intrusion, or to enhance identity assurance with "step-up authentication". Newer intrusion detection technologies can detect anomalies in network events, adding context and risk-based analysis to login attempts and user actions. Correlating network traffic characteristics with normal user behavior may be the key to stopping a "wolf in sheep's clothing".

1. Introduction

Throughout history, authenticating to a computer system was simple: the user provided credentials, the system checked the credentials against a trusted source, and the system permitted or denied access to a protected resource. With technological advances in network monitoring and analysis, this approach can be enhanced to provide identity assurance and context for user activity.

In a previous paper, *Daisy Chain Authentication* (Imbert, 2013), the author explored attacks in which the attacker gathers a collection of accounts using public information and compromised data, building a "daisy chain" to a target. These attacks can be difficult to detect using traditional protection, because the attacker uses normal user login and password recovery procedures to infiltrate the organization. One proposed solution is to collect and analyze data from multiple sources to find login anomalies.

This paper will explore the potential of using network data to increase the level of confidence in the identity of users. The goal is to collect data points and compile them into an identity profile that is helpful in assigning a confidence level to successful and unsuccessful user activities.

In December 2013, OASIS (oasis-open.org) released the first version of a document describing an Electronic Identity Credential Trust Elevation Framework (OASIS, 2013). As defined by OASIS, eCommerce currently uses two models for secure trusted transactions, the credential model and the transaction model.

In the **credential model,** an application validates credentials provided by the user, and permits access to protected resources if the credentials are valid. The credential issuer establishes and validates the trustworthiness of the credential.

In the **transaction model**, an application determines the trustworthiness of the user based on tests against a transaction requested by the user. The application then determines trust and reliability based on a risk model applied against these tests. To the user, the transaction model looks similar to the credential model: he likely still logs on with an assertion of identity, like a username and password. However, transaction models allow for broader definitions of "tests" than the credential model, allowing for the calculation of confidence levels in a user identity.

Courtney Imbert, courtneyimbert@gmail.com

Generally, transaction models would be used by applications, but this paper explores the idea of using data provided by network traffic to expand this approach.

2. Enhancing Level of Assurance (LOA) with Network Analysis

Level of Assurance (LOA) is the degree to which one party is confident that the credential being presented actually represents the owner of those credentials. As Internet transactions become more common and more critical, a high LOA is increasingly important. Meanwhile, attacker capabilities have increased. A common method for infiltrating organizations is to compromise a user's computer or mobile device. Malware can scrape passwords from a browser cache, send keystrokes back to the attacker, or check for password storage.

Additionally, user databases for online services are a popular target for attackers. Studies find 70-80% of users reuse passwords across multiple services (Trusteer, 2010). Compromising a password database is an effective way to compromise users across multiple services.

For critical data, passwords alone do not provide a high enough LOA. Although multi-factor authentication is an effective solution against credential theft, it can be difficult to scale, and it is not foolproof. On the web, a common implementation of multi-factor authentication is to send an SMS message to a mobile device, and require the user to enter the contents of the SMS as well as a secret password. However, if the user stores or caches his password on his mobile device, the two factors of identity are reduced to a single factor – possession of the phone. Whether single or multi-factor, fraudulent use of credentials is a favorite method of breaching information security.

The Trust Elevation Framework provides a list of authentication risk vectors and mitigation strategies for those risks (OASIS, 2013). Internet traffic headers were designed for performance and reliability, not as credentials; as a result, it would be unwise and difficult to rely on network traffic characteristics to identify a user. However, some fields can be used to increase the Level of Assurance of an existing user's credentials, and the Trust Elevation Framework recommends the use of some Internet traffic headers for that purpose. These are shown in Table 1 below. The bolded trust elevation techniques are addressed in this paper.

Courtney Imbert, courtneyimbert@gmail.com

Table 1. Authentication Risk Vectors and Trust Elevation Techniques

Threat	Trust Elevation Techniques
Impersonation Some examples of impersonation are when an entity illegitimately uses another entity's identity information, eg. when a device registers with a network using a spoofed Media Access Control (MAC) address.	Strong AuthN as defined by ITU-T X. 1254 **Per-service device identification** **KBA (time of day)** Biometric **Geolocation**
Online Guessing An attacker performs repeated logon attempts by guessing possible values of the credential.	Physical Biometrics Behavioral Biometrics Hard token Digital certificates KBA with transaction controls Cookie as an additional credential **IP Address** **Time of Access** **Browsing Patterns** **Context**
Credential Theft A device that generates or contains credentials is stolen by an attacker.	Elevate Trust through the use of MFA KBA protected from replay Cookie and **IP Address** Hard token (RSA) Digital certificate protected by password or alternative **Time of Access** **Browsing Patterns** Mouse Patterns **Context**

Source: OASIS, 2013. Retrieved from http://docs.oasis-open.org/trust-el/trust-el-framework/v1.0/trust-el-framework-v1.0.html

Using network traffic to enhance LOA provides several advantages as an information security strategy. The technique is transparent to the user. If an organization does use network analysis to ask for privilege escalation for sensitive or anomalous tasks, it may stop an attacker from using

stolen credentials. Even without a risk engine, a security activity stream can have value if used as an alert system to notify the user of recent and ongoing transactions.

The information used in a network traffic transactional trust model can be used to audit or develop security roles or policies. Traffic analysis can help find valid users violating company policy or working around business policies (for example, multiple users on a single computer, or a single username shared by multiple people).

Malware that uses legitimate user credentials may perform suspicious behavior on the network. If a user with minimal network activity has a computer that becomes compromised by malware, the malware may exhibit unusual behavior like scanning for server shares or issuing continuous Command and Control (C2) traffic. This behavior can sometimes be found by analyzing network traffic, and the ability to detect malicious traffic is further enhanced with anomaly detection.

Trusted entities often use cookies or tokens to identify users. By using network traffic characteristics, no files need be stored on the user's computer, which makes it possible to find identifying data even when users are in private web browser modes or using a variety of devices and software to access systems.

In this paper, a methodology is proposed for using network data to determine the LOA of user identity for requested transactions, using techniques suggested in the Trust Elevation Framework. This methodology consists of four steps, as shown in Figure 1. The steps include collecting data as it crosses the network, analyzing the captured data to determine thresholds for normal and anomalous user behavior, detecting an anomalous series of events based on those thresholds, and evaluating the risk of the correlated events. A higher risk level indicates less confidence in the identity of the user performing the transactions, possibly prompting a responsive action or alert.

Courtney Imbert, courtneyimbert@gmail.com

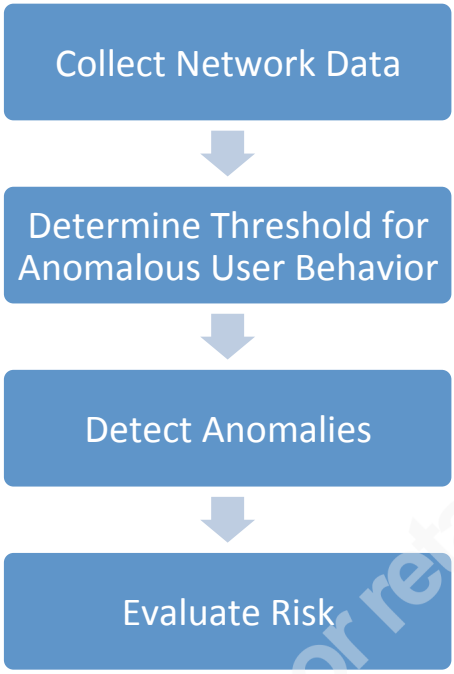

Figure 1. A suggested methodology for increasing LOA with network data.

3. Collecting Network Data to Enhance Identity Assurance – A Case Study

In this section, a case study will be presented which will demonstrate the collection and analysis of identifying information provided by network traffic captures. The usage of the tools which will be described during this case study will just demonstrate proof-of-concept and cannot be considered a practical, large-scale method of identifying users on a network.

In the examples contained within this case study, credentials and identifying information will be extracted from the Internet, Transport, and Application layers of sample packets.

3.1 Creating the sample data

The case study used a simple virtual network with several workstations, a network traffic collector, and one server. Five users exist in the network: Dave, Wendy, Bernard, Sandy, and Jeff. The network has a mix of Windows 7 and Linux virtual machines. Bernard and Sandy share a computer, but have unique data usage.

The central CentOS server hosts HTTP, FTP and Telnet services (all unencrypted protocols). Usage was simulated across multiple workstations and accounts. It is worth noting that the services on the network contain unencrypted traffic for demonstration purposes, but in a production environment, encrypted alternatives like HTTPS and SSH are preferable. Analysis of encrypted traffic is briefly explored in this paper as a challenge when searching network traffic for user-specific information.

tcpdump was used to capture and filter packets crossing the network. tcpdump is a versatile command-line packet collector and analyzer. It is available for download at http://www.tcpdump.org.

tcprewrite is a tool in the Tcpreplay suite, a group of licensed tools that provide the ability to modify and replay previously captured traffic. The tool is available at http://tcpreplay.synfin.net . In the virtual lab environment, users shared a subnet. To demonstrate the geolocation component, random IP addresses were generated using tcprewrite and the `--seed` option, then the source IP addresses in the original capture file were replaced using the `--pnat` option. The results of the randomization and substitution process are shown in Table 2.

Table 2. IP Address Substitution for Sample Traffic

User	Original IP Address	Random IP Address
bernard sandy	192.168.80.154	75.172.209.151
dave	192.168.80.145	94.239.121.157
wendy	192.168.80.156	126.157.219.153
server	192.168.80.157	78.187.185.183

After testing open-source tools, NetworkMiner was selected for data analysis due to its versatility and interface. NetworkMiner is a network forensics tool for analyzing captured network traffic, and it includes a variety of analysis capabilities with a GUI interface. NetworkMiner uses Satori and p0f databases for fingerprinting. Wireshark was used for features that the free version of NetworkMiner did not support. NetworkMiner is free and open-source, though the premium version contains additional features. NetworkMiner is available for download at http://sourceforge.net/projects/networkminer/ .

Courtney Imbert, courtneyimbert@gmail.com

iptables was used to write and test firewall rules. iptables is a command line program that allows a system administrator to configure the Linux packet filtering ruleset. It can be downloaded from http://www.netfilter.org/projects/iptables/ .

Snort was used to demonstrate IDS/IPS rules. Snort is a network-based intrusion prevention system (NIPS) that performs real-time traffic analysis and packet logging. Snort can be downloaded from https://www.snort.org/ .

3.2 Extracting Credentials from Network Traffic Using NetworkMiner

After capturing traffic from simulated usage, NetworkMiner was used to analyze the network traffic and extract credentials. Shown in Figure 2 are the results of NetworkMiner analysis of the sample network traffic.

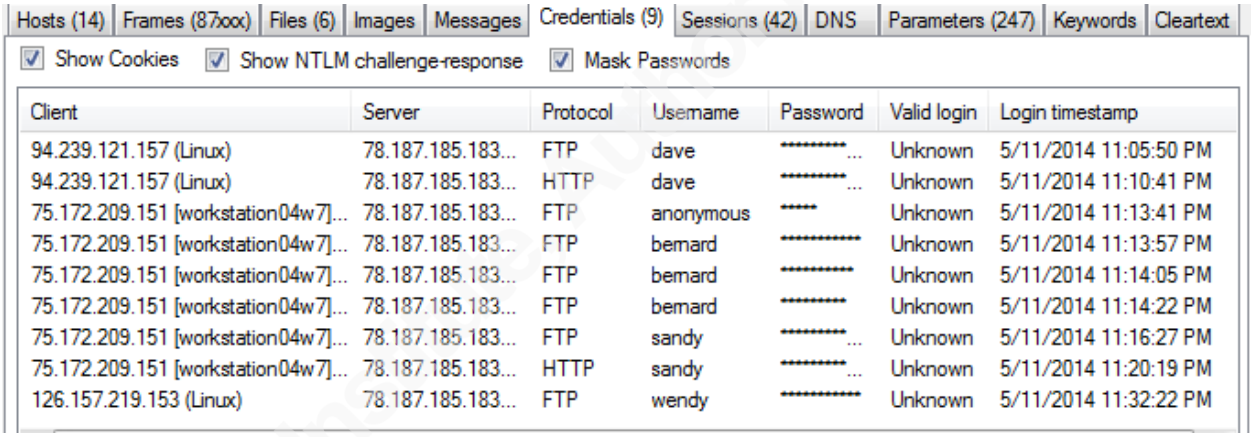

Figure 2. NetworkMiner credential results from a scan of a sample network traffic capture.

The NetworkMiner analyzer simply recognizes credentials crossing the network, and does not determine if the attempts are successful. As a result, Bernard's unsuccessful password attempts are included in this capture, as well as an unsuccessful anonymous login. To view only successful logins, it would be more efficient to collect data from application logs.

Not all login activity is included here; for example, several users initiated Telnet connections into 78.187.185.183, but NetworkMiner did not recognize them. Telnet logins, though unencrypted, are fragmented into a single character per packet, and unsupported by NetworkMiner (Netresec, 2014). Additionally, credentials from encrypted protocols like SSH would not be extracted by this tool.

Courtney Imbert, courtneyimbert@gmail.com

3.3 Extracting Credentials Using a Sample of Traffic

Extracting credentials from network traffic can be complex due to the large number of possible protocols and the ways they handle credentials. It is possible to extract credentials by analyzing the fields in a representative sample of traffic, then searching for similar characteristics (Davidoff & Ham, 2012). The following demonstration uses FTP as a sample protocol for this process.

By using the "prepare filter" functionality of Wireshark, the User ID field in FTP is determined to be `ftp.request.command == "USER"`, as shown in Figure 3.

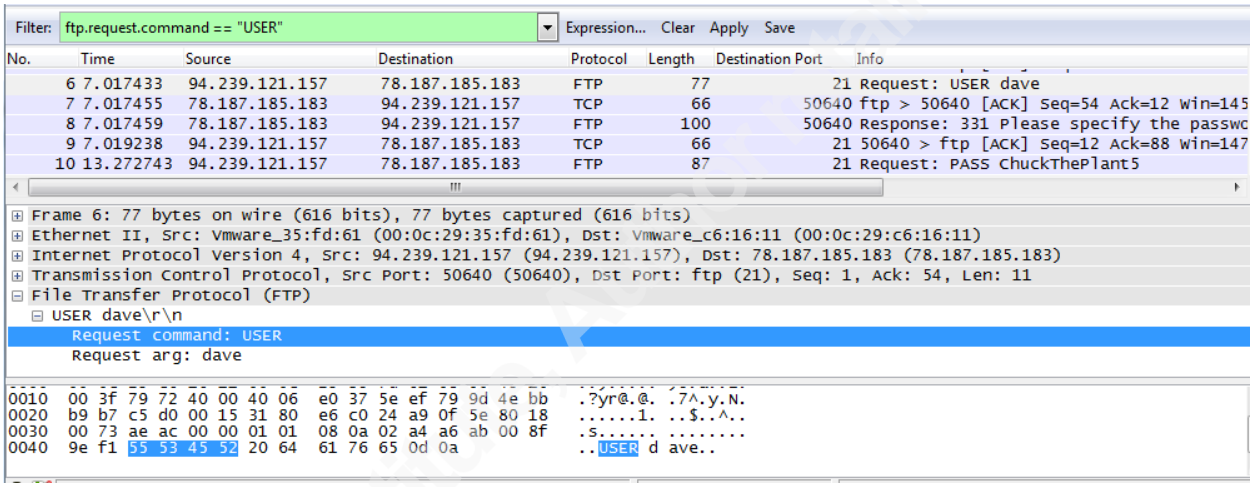

Figure 3. Locating the Wireshark field name using the contents of a sample packet.

Using the same technique, the fields and associated values of a successful FTP login were compiled into Table 3 shown below.

Table 3. Field Names & Values for Successful FTP Login Attempts

Packet contents (Wireshark / tshark filter)	Description
`ftp.response.code == 230`	Indicates a successful login
`ftp.request.command == "USER"` `ftp.request.arg == "<UserID>"`	Indicates a user ID request User ID response

Courtney Imbert, courtneyimbert@gmail.com

Extraction of usernames can be accomplished using a tool like tshark, with some understanding of the protocol used to log in. For example, using this information, the following filter captures only successful logins and the associated credentials:

```
tshark –r capture.pcap –w ftpCredentials.pcap –Y
"(ftp.response.code == 230 || ftp.request.command == "USER")"
```

Once that is complete, the user ID field can be extracted from the file using the –e flag:

```
tshark –r ftpCredentials.pcap –T fields –e ftp.request.command –
e ftp.request.arg
```

This command returns an output that looks similar to this:

```
USER dave
USER bernard
USER sandy
USER wendy
```

Without the help of an automated tool, this process would need to be completed for each protocol in use on the network, and would require some decoding of credentials; or, in case of encryption, it may yield few useful results.

3.4 Source Address Profiling

The most obvious information to begin compiling in a user profile based on network traffic is the source IP address. Table 4 shows an initial list of user profiles consisting of username, IP address, and the services used (as determined by protocol/port).

Table 4. Partial User Profiles, including Services / Protocols by User

User	IP Address	Services Used
bernard	75.172.209.151	FTP
dave	94.239.121.157	FTP, HTTP
wendy	126.157.219.153	FTP
sandy	75.172.209.151	FTP, HTTP

For sensitive usernames that are expected to be accessed only from specific networks or hosts, Snort or IDS/IPS rules can be used. For example, the following Snort rule alerts on ftp connections to the server located at 78.187.185.183 that contain user ID "bernard" if they did not

originate from the 75.172.209.0/24 subnet, using Perl Compatible Regular Expressions to detect the word “user”, followed by whitespace, followed by the word “bernard”:

```
alert tcp !75.172.209.0/24 any -> 78.187.185.183 21 (msg:”ftp
bernard login from possible unauthorized host”;
flow:to_server,established; content:”bernard”;
pcre:”/user\s+bernard”;)
```

For networks over which the analyst has control, IP addresses and subnets may be easier to correlate with individual users. However, some technologies may obscure IP source addresses, making profiling less effective. Examples include network address translation, perimeter protection, onion routing, and encryption or tunneling (Bejtlich, 2013).

3.5 Geolocation

Geolocation by IP address is a common technique for determining the location of a user, but not the most accurate. However, if it is the only clue available, it can be used to determine a rough location.

As of the writing of this paper, NetworkMiner includes IP address geolocation only in the premium version (Netresec, 2014). Several options exist for determining the geolocation of IP addresses. MaxMind’s GeoLite, an IP intelligence product designed to geolocate hosts, integrates with Wireshark, and supports IPv4 and IPv6. MaxMind is available for download at http://dev.maxmind.com/geoip/ .

Figure 4 shows the Statistics / Endpoints report of Wireshark against the sample capture, and includes geolocation based on a GeoLite lookup from captured IP addresses.

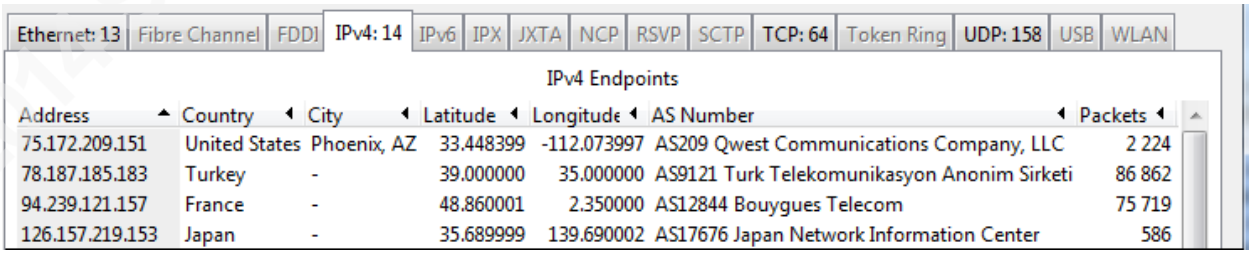

Figure 4. Wireshark Geolocation Statistics by IP Address against sample data.

Wireshark will also generate a map of the locations found in the packet capture, as shown in Figure 5.

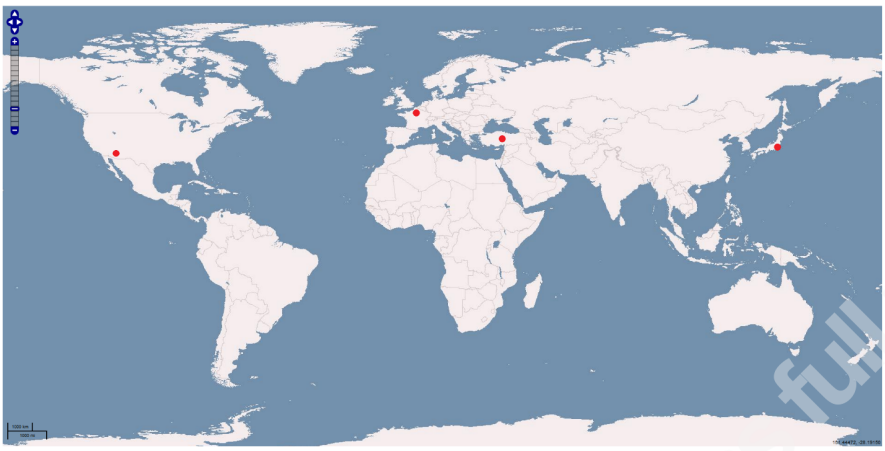

Figure 5. Wireshark map interface, including the location of hosts from sample data.

The rough geolocation for each IP address was added to each user profile as shown in Table 5.

Table 5. Partial User-based Profiles, including Geolocation.

User	IP Address	Country	Services Used
bernard	75.172.209.151	USA	FTP
dave	94.239.121.157	France	FTP, HTTP
wendy	126.157.219.153	Japan	FTP
sandy	75.172.209.151	USA	FTP, HTTP

Many other techniques exist within applications to determine geolocation. Newer browsers permit the use of the W3C Geolocation standard (Popescu, 2012), which is supported by HTML 5. The standard allows for detecting geolocation with more accurate methods than IP address, including GPS sensors. Using the W3C geolocation standard to determine location would require that network users access a website that uses the W3C Geolocation API, and the user must permit the site to use his geolocation (Tomes & O'Connor, 2013).

3.6 Device OS and Browser Fingerprinting with p0f

p0f is a fingerprinting tool that can be used to identify characteristics like operating systems and applications installed on devices on a network. It guesses the operating system of the hosts by examining the header and contents of network traffic, and comparing these characteristics to a database of operating system and application characteristics. The tool can also fingerprint browsers and other applications by gathering content from the packet payloads. p0f is available for download from http://lcamtuf.coredump.cx/p0f3/ .

Courtney Imbert, courtneyimbert@gmail.com

NetworkMiner uses Satori and p0f databases to extract OS and browser fingerprints from a packet capture. Passive fingerprinting is not a precise science; it can use only the data that happens to pass the network, which may not be enough for certainty. As shown in Figure 6 below, p0f identified the machine at 75.172.209.151 only as a Windows computer, while Satori assigned percentage values to the level of confidence for each operating system (all Windows). In reality, this host has Windows 7 installed.

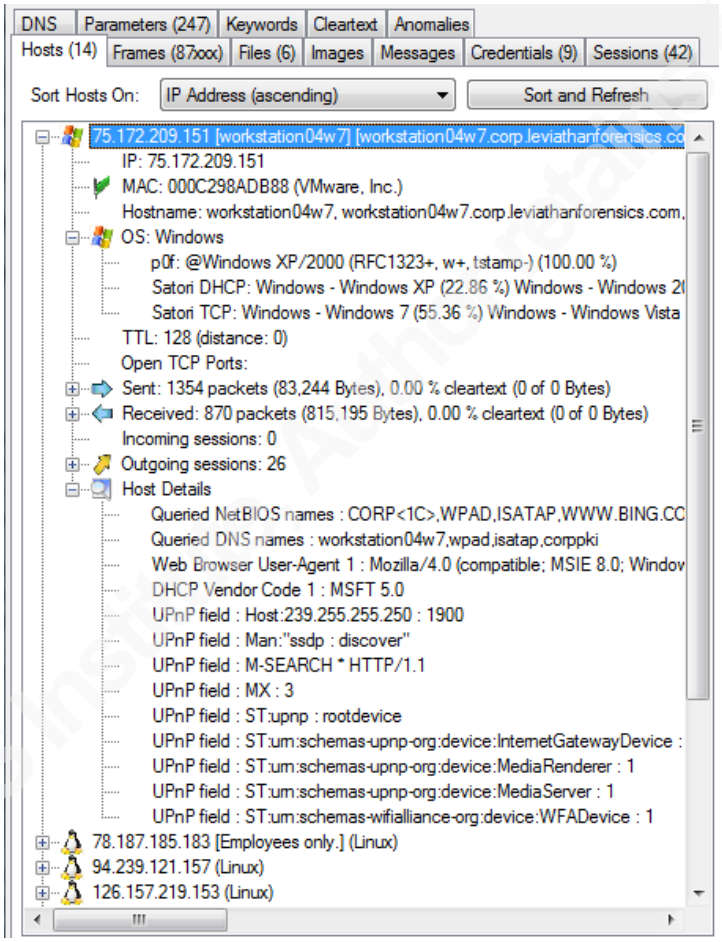

Figure 6. NetworkMiner detailed information for host workstation04w7.

By correlating the fingerprint data with the source IP addresses identified by extracting credentials, more detail can be added, as shown in Table 6.

Courtney Imbert, courtneyimbert@gmail.com

Table 6. Partial User-based Profiles, including OS/Browser Fingerprinting.

User	IP Address	Country	Hostname	OS	Browser	Services Used
bernard	75.172.209.151	USA	workstation04w7	Windows	MSIE	FTP
dave	94.239.121.157	France	unknown	Linux	Firefox	FTP, HTTP
wendy	126.157.219.153	Japan	unknown	Linux	unknown	FTP
sandy	75.172.209.151	USA	workstation04w7	Windows	Firefox	FTP, HTTP

Techniques that make OS and browser fingerprinting less effective include user-agent masking, browser features, users switching between browsers, software patching/updates, proxy servers that change traffic characteristics, and encryption or tunneling.

3.7 Time of day and login frequency

In a global business environment where users expect access to network resources 24/7, limiting network access by time of day is increasingly impractical. However, if users are limited to a single time zone and sensitive tasks occur only during business hours, it may be helpful to profile and/or limit users by the time of day they access the network.

Gathering data on the time of day and login frequency based on network traffic requires traffic captures over a longer period of time. Due to the large volume of data collected by deep packet inspection, it may be more effective to perform this type of analysis by inspecting only packet headers, resulting in a profile by host rather than by user.

Packet captures include timestamps, which make it simple to make a graph of network usage. Wireshark's IO Graph, shown below in Figure 7, can provide a graphical representation of access by day, using the number of packets or the volume of data transferred.

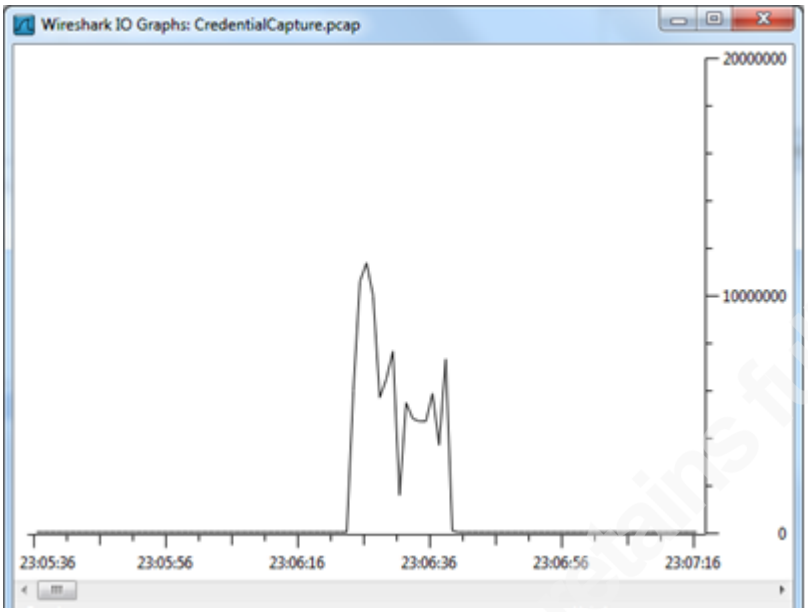

Figure 7. Wireshark IO Graph of Network Traffic Volume by Time.

Unfortunately, Snort does not have the capability to alert on events that occur only at specific times of day. However, iptables firewall rules can limit access to hosts or networks at specific times. (Singh, 2012)

If a user has never logged in to an account, or rarely does, an attempt to log in may be a higher-risk indicator than a similar login attempt for a frequent user. Astute readers will notice Jeff, one of the sample users, had no detected logins on the network during the sample traffic capture. Any login from Jeff may warrant additional investigation; the account may be inactive, in which case it should be disabled. Jeff may simply be on vacation, or he may use a method of accessing data that hides network visibility to his login attempts.

3.8 Traffic Flow indicators

Network profiling and anomaly detection have become more sophisticated with the introduction of network flow analysis. Part of the behavioral profile of a user includes the typical volume, source, and destination of traffic. These components are included in network flow analysis – analyzing not individual packets, but the overall flow from one point to another. Because the volume of collected data is lower than deep packet inspection, traffic flow analysis is ideal for high-speed or high-volume networks (Muraleedharan, 2009) . This approach helps detect anomalous traffic patterns, even if the traffic is encrypted.

Courtney Imbert, courtneyimbert@gmail.com

SiLK is a suite of network traffic collection and analysis tools designed for security analysis of traffic flows in large networks. It is available for download at https://tools.netsa.cert.org/silk/download.html . SiLK flows are defined by five attributes: source IP address, destination IP address, source port, destination port, and transport protocol. The flows themselves contain information about the source address, source port, destination port, IP protocol, the number of bytes, accumulated TCP flags, start and end time, sensor identity, flow termination conditions, and application-layer protocol (Collins, 2014).

SiLK is a robust tool, and its full range of usage is outside the scope of this paper. However, it is helpful to demonstrate traffic flows using SiLK against the sample network capture.

YAF (http://tools.netsa.cert.org/yaf/), a tool that processes packet data from pcap files into bidirectional flows, was used to convert the collected network data into network flows SiLK could process.

```
rwp2yaf2silk --in=CredentialCapture.pcap --
out=CredentialCapture.silk
```

For the purpose of profiling users, rwstats was run against the SiLK-compatible file generated by YAF to report on total bytes downloaded and uploaded by each host.

Downloads / Received traffic:

```
rwstats CredentialCapture.silk --fields=dip --values=bytes --
count 10 > downloaders.txt
```

```
INPUT: 310 Records for 14 Bins and 109110928 Total Bytes
OUTPUT: Top 10 Bins by Bytes

            dIP|               Bytes|    %Bytes|   cumul_%|
94.239.121.157|            94245295| 86.375670| 86.375670|
   192.168.80.1|            13266541| 12.158765| 98.534435|
75.172.209.151|              815195|  0.747125| 99.281560|
78.187.185.183|              711882|  0.652439| 99.933998|
192.168.80.255|               34255|  0.031395| 99.965393|
   224.0.0.252|               12554|  0.011506| 99.976899|
126.157.219.153|              12130|  0.011117| 99.988016|
239.255.255.250|              5316|  0.004872| 99.992888|
```

Courtney Imbert, courtneyimbert@gmail.com

```
255.255.255.255|                2352|   0.002156| 99.995044|
192.168.80.254|                2010|   0.001842| 99.996886|
```

Uploads / Sent traffic:

```
rwstats CredentialCapture.silk -fields=sip --values=bytes --
count 7 > uploaders.txt

INPUT: 310 Records for 7 Bins and 109110928 Total Bytes
OUTPUT: Top 10 Bins by Bytes
           sIP|               Bytes|    %Bytes|   cumul_%|
78.187.185.183|           108334017| 99.287962| 99.287962|
94.239.121.157|              663226|  0.607846| 99.895808|
75.172.209.151|               83244|  0.076293| 99.972101|
126.157.219.153|              19651|  0.018010| 99.990111|
192.168.80.155|               5822|  0.005336| 99.995447|
192.168.80.254|               4312|  0.003952| 99.999399|
192.168.80.138|                656|  0.000601|100.000000|
```

The average download volume was calculated and included in the profile, as shown in Table 7. The sample network capture was taken over the course of 29 minutes, so the number used on the profile is the average number of bytes per second during the network capture.

Table 7. User-based Profiles, including Average Rate of Network Usage.

User	IP Address	Country	Hostname	OS	Browser	Services Used	Bytes/sec (down)	Bytes/sec (up)
bernard	75.172.209.151	USA	workstation04w7	Windows	MSIE	FTP	469	48
dave	94.239.121.157	France	unknown	Linux	Firefox	FTP, HTTP	54164	381
wendy	126.157.219.153	Japan	unknown	Linux	unknown	FTP	7	11
sandy	75.172.209.151	USA	workstation04w7	Windows	Firefox	FTP, HTTP	469	48

As shown in Table 7, Dave is the heaviest user of network resources for both up- and downloading. Wendy is a light user. Because network flows use only the headers of network traffic for analysis, Sandy and Bernard (who share a workstation and IP address) have the same statistics – an illustration of the problem of using just packet headers when profiling individual user behavior.

Flow-based traffic indicators have one benefit that device profiles do not provide: they can identify anomalous *behavior*. If a machine is compromised, including compromise of the

associated username, any traffic originating from that machine will match normal device characteristics. If a trusted device is compromised, traffic flow monitoring can catch malware behavior like exfiltration or scanning.

4. Determining Thresholds for Anomalous User Behavior

When using thresholds in anomaly detection, attributes of user and system behavior are counted, with some level established as permissible for each of those attributes. When the threshold for suspicious attributes is exceeded, the behavior is considered abnormal and an alert is generated.

Thresholds may be reached in absolute terms, but commonly, they are measured and alerted upon for data over a specified time period. For example, an alert may occur when a single user downloads over ten gigabytes of data over a two-hour time period.

Shown in Table 8 is a collection of traffic volume from user "dave" at IP address 94.239.121.157, measured in five 5-minute slices. This is a demonstration of a quantitative statistical anomaly threshold calculation for one user, using a very small sample for simplicity.

Table 8. Traffic Volume from user "dave"

Start time	End time	Up (bytes)	Down (bytes)
23:01	23:05	254835	41553
23:06	23:10	58845136	15684794
23:11	23:15	1586587	48775630
23:16	23:20	34564804	18875782
23:21	23:25	13082655	10867536

Once the data is collected, the upstream and downstream traffic can be calculated against as shown in Table 9 below: the number of samples, the mean, and the sample standard deviation.

Table 9. Calculations based on traffic collected from user "dave"

Calculation	Upstream	Downstream
N	5	5
Total volume	108334017	94245295
Mean volume	21666803	18849059
Sample Std. Dev.	24919681	18186808

Using this data, traffic from Dave's IP address might be flagged if it exceeds, for example, two standard deviations from his mean traffic - 71506164 bytes up or 55222675 bytes down over the span of one hour.

Calculating a mean and standard deviation of collected numerical values is one of the simplest and most common ways to determine thresholds, but there are many others. The challenge of setting anomaly thresholds is to find a method that delivers an accurate threshold - one that does not generate excessive false positives or negatives. "Normal" network traffic varies from organization to organization, and from user to user. Therefore, rule and threshold development benefits from collaboration from people with an understanding of the ebbs and flows of business activity.

Table 10 presents some examples of business-related questions and impacts the answers may have on network traffic thresholds or rules.

Courtney Imbert, courtneyimbert@gmail.com

Table 10. Possible Questions and Resulting Thresholds or Rules.

Question	Answer may lead to thresholds / rules based on…
Are users expected to use a single work computer to access resources, or do they bring their own devices?	Limiting host or user access to network resources or protocols/services
Do users have control over the browser or applications they use to access network resources?	Limiting host or application access to network resources (most likely done via application, not at the network level)
Do users have widely varying network usage over time, or is their network and computer usage highly stable?	Download traffic thresholds over time (erratic network usage would benefit from a broader threshold before flagging traffic)

5. Anomaly Detection Techniques

Currently, intrusion detection can be broadly classified into at least two methodologies: **signature-based** detection (sometimes called **misuse detection**), and **anomaly-based detection**.

Signature detection searches for activity that matches known signatures of intrusions, and normally uses rules or signatures to identify those known attacks.

Anomaly-based detection detects attacks that include behavior outside the historic norm, meaning it has the ability to identify previously unidentified attack types. Because anomaly detection often uses a "learning period" to collect profiles of normal behavior and then analyzes current behavior against the profile, it is compatible with identifying unusual user behavior (Scarfone & Mell, 2007). Since attackers with stolen credentials often exhibit different behavior from the true owner of the credentials, anomaly detection techniques can be used to provide assurance in a user's identity.

5.1 Rule-based Anomaly Detection

In a rule-based technique for anomaly detection, observed data defines rules for acceptable usage patterns. The difference between rule-based anomaly detection and signature-based (misuse) detection is that rule-based anomaly detection uses historical data to determine whether an attack is taking place, while signature-based detection does not consider the historical context of the

events.

The drawback of rule-based anomaly detection is that a rule set must be defined, which can be work-intensive to create and maintain. As signatures or rules increase, performance decreases, and enforcing rules that require deep packet inspection also has a negative impact on performance. Therefore, rule-based intrusion analysis are most appropriate for rules and policies that are quickly processed and universal for the organization (Sen, 2007).

5.2 Statistical Anomaly Detection

In statistical anomaly detection, behavior is measured by specific variables sampled over time. The variables are stored and analyzed, and the results are maintained in a profile. The current behavior of each user is compared with the stored profile, and any system event that deviates from the expected profile by some defined value is flagged as an intrusion attempt. Statistical anomaly detection is most appropriate when the collected data is quantitative and has real values (Sarasamma & Huff, 2005).

5.3 Non-Linear Measures ("Soft computing")

Artificial intelligence (AI) is a relatively new technique for network intrusion detection. AI uses learning and induction to find anomalies, perform discoveries, and improve performance (Russell & Norvig, 1995). It has obvious benefits in the complex and ever-changing world of computer networks. Currently, many IDSes employ AI methods in their systems to reduce data sets, and to classify network traffic (Kumar, 2012).

5.3.1 Artificial Neural networks

An artificial Neural Network is an algorithm that gathers inputs from a sensor, then transforms the data to a set of searched outputs through a set of processing units, or through a series of nodes and connections. Neural networks are computationally intensive and still under development for commercial use, but a possibility for the future of intrusion analysis (Reddy, 2013).

5.3.2 Genetic algorithms

A genetic algorithm (GA) mimics biological evolution as a problem-solving strategy. The goal is to optimize a population of candidate solutions toward a predetermined fitness. Like neural

networks, genetic algorithms are relatively new, and still under development for commercial intrusion detection solutions (Hoque, Mukit, & Bikas, 2012).

6. Risk Engines

One of the general challenges of intrusion detection lies in finding the context of a single event. Individual alerts may not seem like a significant problem, but when combined, they may indicate an attack. For example, consider the anomalous events shown in Table 11.

Table 11. Example Events and Possible IDS Flags

Event	Why flagged
8 unsuccessful login attempts against username "wendy" on FTP server 78.187.185.183	An unsuccessful login attempt may indicate an attempt at user compromise
User "wendy" successfully logs into server 78.187.185.183 from IP address 75.172.209.151	"wendy" has a workstation at IP address 126.157.219.153, which does not match the source IP address
1 GB of network traffic passes from IP address 78.187.185.183 to an Internet IP address over a short time period	Abnormally high upload traffic

Individually, these events may not be significant enough to arouse suspicion or warrant preventative action. However, when pieced together, they hint at password guessing attempts against the "wendy" account, followed by data exfiltration from the FTP server to an external destination.

A risk engine uses a heuristic analysis instead of a signature-based one. Heuristics often have a tradeoff - accuracy, precision, or completeness for speed. Though heuristic approaches have long been used for anti-malware solutions, they are also applicable to network security. A risk engine collects transactional data from multiple channels, assigns a risk factor to each, and compiles a threat level based on the combined data over a period of time (Baylor, 2013). If the threat level exceeds the threshold set by the organization, the network can initiate an alert or preventative action.

To calculate the threat level of a series of events, the risk engine must set boundaries around the data. For the sake of detecting anomalous logins, this may include any combination of any of the fields we gathered when profiling users.

Recall the values in wendy's profile:

```
Username: wendy
IP Address: 126.157.219.153
Country: Japan
Hostname: unknown
OS: Linux
Browser: unknown
Services/Protocols used: FTP
Average Bytes/sec down: 7
Average Bytes/sec up: 11
```

It might be useful to collect any events that are unique to a user into a correlated series; in this case, events that include packets with username "wendy" or an IP address of 126.157.219.153. In order to correlate the traffic from the large data transfer, the risk engine would need to "follow" Wendy. That is, the risk engine would need to be aware of Wendy's *successful* login to 78.187.185.183, and begin correlating events from that session with the rest of the series. When the network traffic sensor detects that the user has logged out, the risk engine should stop using that IP address or session as an identifier. A possible sequence of events is shown below in Table 12, with the correlation triggers highlighted. The risk rating for each event type was chosen arbitrarily.

Table 12. Correlated Network Events from User "wendy"

Time	Event	Username	Src IP	Dst IP	Risk Rating	Action
1:00	Unsuccessful FTP login	wendy	75.172.209.151	78.187.185.183	1	
1:01	Unsuccessful FTP login	wendy	75.172.209.151	78.187.185.183	1	
1:02	Unsuccessful FTP login	wendy	75.172.209.151	78.187.185.183	1	
1:03	Unsuccessful FTP login	wendy	75.172.209.151	78.187.185.183	1	
1:04	Unsuccessful FTP login	wendy	75.172.209.151	78.187.185.183	1	
1:05	Unsuccessful FTP login	wendy	75.172.209.151	78.187.185.183	1	
1:06	Unsuccessful FTP login	wendy	75.172.209.151	78.187.185.183	1	
1:07	Unsuccessful FTP login	wendy	75.172.209.151	78.187.185.183	1	
1:08	Successful FTP login	wendy	75.172.209.151	78.187.185.183	0	Begin including events from IP address 78.187.185.183
1:09	High levels of traffic	unknown	78.187.185.183	198.51.100.101	3	
1:15	FTP connection closed	wendy	75.172.209.151	78.187.185.183	0	Stop including events from IP address 78.187.185.183

Courtney Imbert, courtneyimbert@gmail.com

Based on the results of this correlation, the cumulative risk rating is 11 for a series of actions identified as potentially belonging to user "wendy". If the organization's threshold for dangerously low assurance of identity is a risk rating of 10 over the span of 15 minutes, the risk engine would log, alert, or otherwise initiate action when that threshold is exceeded at 1:09.

This policy could facilitate the Trust Elevation model; for example, if the confidence factor is low, the risk engine could require an application to provide additional proof of identity, or simply deny access to a high-risk transaction.

Using historical data or the user profile to detect anomalies would further enhance this technique, as shown in Table 13. Anomalies against the user's historical profile might result in a higher risk rating.

Courtney Imbert, courtneyimbert@gmail.com

Table 13. Correlated Network Events from User "wendy", including Anomalies

Time	Event	Username	Src IP	Dst IP	Risk Rating	Anomaly	Action
1:00	Unsuccessful FTP login	wendy	75.172.209.151	78.187.185.183	1	None (unsuccessful logins = 1)	
1:01	Unsuccessful FTP login	wendy	75.172.209.151	78.187.185.183	1	None (unsuccessful logins = 2)	
1:02	Unsuccessful FTP login	wendy	75.172.209.151	78.187.185.183	1	None (unsuccessful logins = 3)	
1:03	Unsuccessful FTP login	wendy	75.172.209.151	78.187.185.183	1	None (unsuccessful logins = 4)	
1:04	Unsuccessful FTP login	wendy	75.172.209.151	78.187.185.183	1	None (unsuccessful logins = 5)	
1:05	Unsuccessful FTP login	wendy	75.172.209.151	78.187.185.183	2	2 std. dev. above mean login attempts	Increase risk rating due to anomaly
1:06	Unsuccessful FTP login	wendy	75.172.209.151	78.187.185.183	2	2 std. dev. above mean login attempts	Increase risk rating as number of unsuccessful logins increase
1:07	Unsuccessful FTP login	wendy	75.172.209.151	78.187.185.183	2	2 std. dev. above mean login attempts	Increase risk rating as number of unsuccessful logins increase
1:08	Successful FTP login	wendy	75.172.209.151	78.187.185.183	2	The IP address in wendy's profile does not match the source address	Increase risk rating due to anomaly. Begin including events from IP address 78.187.185.183
1:09	High levels of traffic	unknown	78.187.185.183	198.51.100.101	4	2 std. dev. above the mean traffic rate in wendy's profile	Increase risk rating due to anomaly.
1:15	FTP connection closed	wendy	75.172.209.151	78.187.185.183	0		Stop including events from IP address 78.187.185.183

This is a thought exercise on how a risk engine might correlate network events with individual users, adapt to anomalies, and compute the total risk of a cluster of events. Practically, a tool like Bro could be used to implement such a technique. Bro has scripting capabilities that can detect multi-stage attacks, as well as collect and reference historical information or profiles (Runnels, 2012). Though an exploration of the Bro scripting language is outside the scope of this paper, the capabilities of Bro show promise for anomaly detection and adaptive response to events (Sommer, 2007). For example, shown below is a snippet of a Bro script that collects the number

of unsuccessful FTP logins using the FTP reply code, tracks the originating host ($c$id$orig_h), and acts upon the cumulative number of failed attempts.

```
##! Title: detect-bruteforcing.bro
##! Author: The Bro Project
##! Code version date: 2013
##! Availability: http://www.bro.org/sphinx-git/broids/index.html

event ftp_reply(c: connection, code: count, msg: string,
cont_resp: bool)
     {
     local cmd = c$ftp$cmdarg$cmd;
     if ( cmd == "USER" || cmd == "PASS" )
          {
          if ( FTP::parse_ftp_reply_code(code)$x == 5 )
               SumStats::observe("ftp.failed_auth",
[$host=c$id$orig_h], [$str=cat(c$id$resp_h)]);
          }
     }
```

With respect to performance, a user-specific risk engine technique would be difficult to implement due to the computational expense for each event. Each event must be correlated, checked against rules or scripts and the user's profile for anomalies, and added to the total risk rating.

Determining the risk of combined behaviors is complex, and development is still underway to apply the concept to practical solutions. Though no single open-source or free information security tool can perform all the steps, several tools can be pieced together to apply the proposed methodology. Some possible tools are suggested in Appendix A. Commercial information security solutions are beginning to combine user profiling with intrusion detection and risk analysis, and several of these are summarized in Appendix B.

Though implementing a network risk engine presents unique challenges, the risk engine model has the potential to combine login attempts, network traffic characteristics, and transactions when calculating threat levels, making it a powerful tool for detecting anomalous behavior.

Courtney Imbert, courtneyimbert@gmail.com

7. Limitations and challenges to user-based network profiling

7.1 Performance

To identify and perform network analysis of individual users, deep packet inspection is normally necessary. Deep packet inspection may be too computationally expensive for large or complex networks, and although rules with content searching are possible to write, running a large rule set or performance-intensive deep packet inspection against real-time network traffic may not be feasible.

7.2 Encrypted and Compressed Traffic

The simplest credentials to extract from network traffic are unencrypted, unencoded, with a standard login method, and are contained within a single packet. However, the security risks of running unencrypted network services far outweigh the benefits of monitoring individual logins. Increasingly, network traffic is protected or encrypted in some form. This makes it much more difficult to extract credentials from network traffic. Traffic can be encrypted by individual applications, but host or user profiling becomes even more challenging when all network traffic is encrypted, e.g. VPN or onion routing.

There are several options that accommodate encrypted traffic when performing traffic analysis:

Collect and analyze metadata from the encrypted traffic. Though it may not be possible to collect usernames and transactional data from encrypted network traffic, the network flow can be used to develop a profile. This includes the source and destination IP addresses, and the approximate volume of traffic exchanged between hosts. In some cases, network traffic has characteristics that don't require the content of the packet to identify; for example, small packets every 20 milliseconds are characteristic of VoIP traffic (Cisco, 2001).

Position network sensors to intercept requests for encrypted transactions. Though "man in the middle" is usually considered a technique for attackers, it can be used to intercept requests for an encrypted transaction. When the sensor receives a request, it requests a certificate or key from the intended destination, uses its own encryption for communication between the client and sensor, and encrypts/decrypts all traffic before passing it on. This technique is most commonly used for SSL inspection. For an organization that uses VPN to protect clients outside the physical

network, inspection can occur immediately after the traffic is decrypted upon entering the network, and before the traffic is encrypted to exit the network.

Collect and correlate data from multiple sources. If the administrator has visibility into the applications for which he is interested in profiling users, he can use application logs to provide additional context to network traffic. Identity management solutions can even be used to supplement or manage user profile information and create intrusion detection rules or thresholds. With access to supplemental resources, a profile based on network traffic can be compiled with more detail and a higher degree of certainty.

7.3 Shared credentials or devices

As shown in the case study, Sandy and Bernard have unique usage patterns, but they are barely distinguishable in their profile. Most traffic flow analysis is limited, resulting in profiles based on the IP address and common software fingerprint of any users on the computer. This can be overcome by giving users unique devices or network sessions or by using correlated data from another source to differentiate users, like application or event logs.

8. Conclusion

The highly individualized nature of network traffic and multiple authentication systems can make identity management complex, but understanding the context of a single identity performing multiple actions within a network is increasingly necessary to prevent and detect security incidents. Much of the design of network traffic was not intended to identify or authenticate users, but it can be collected, clustered into profiles, and compared with new events to increase the Level of Assurance (LOA) of a user identity. IDS/IPS anomaly detection is ideal for this purpose, resulting in a more holistic view of the network and users alike. Additionally, a risk engine can cluster related and recent activities and calculate a cumulative risk level for that series of events, resulting smarter information security decisions. The technologies to perform the required data correlation are still evolving, but monitoring network traffic and integrating it into identity profiles is an effective method to detect impostors and malicious behavior.

Courtney Imbert, courtneyimbert@gmail.com

References

Baylor, K. (2013, September 17). Risk Engines: Criminally Induced Revolution [Web log post]. Retrieved from https://www.nsslabs.com/blog/risk-engines-criminally-induced-revolution

Bejtlich, R. (2013). *The Practice of Network Security Monitoring.* San Francisco, CA: No Starch Press.

Cisco. (2001, June 30). *Quality of Service for Voice Over IP.* Retrieved from http://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/qos_solutions/QoSVoIP/QoSVoIP.pdf

Collins, M. (2014). *Network Security Through Data Analysis.* Sebastopol, CA: O'Reilly.

Davidoff, S., & Ham, J. (2012). *Network Forensics: Tracking Hackers Through Cyberspace.* Westford, MA: Prentice Hall.

Hoque, M. S., Mukit, M., & Bikas, M. (2012, March). An Implementation of Intrusion Detection System Using Genetic Algorithm. *International Journal of Network Security & Its Applications (IJNSA), Vol. 4, No. 2.* Retrieved from http://arxiv.org/ftp/arxiv/papers/1204/1204.1336.pdf

HP ArcSight. (2012, May 1). HP ArcSight IdentityView [Data sheet]. Retrieved from h20195.www2.hp.com/V2/GetDocument.aspx?docname=4AA3-8958ENW

Imbert, Courtney (2013, June 19). *Daisy Chain Authentication*. Retrieved from http://www.sans.org/reading-room/whitepapers/authentication/daisy-chain-authentication-34292

Kumar, G., & Kumar, K. (2012). The Use of Artificial-Intelligence-Based Ensembles for Intrusion Detection: A Review. *Applied Computational Intelligence and Soft Computing*, Volume 2012, Article ID 850160. Retrieved from http://dx.doi.org/10.1155/2012/850160

Muraleedharan, N. (2009, February). *Flow Based Traffic Analysis.* Retrieved from http://www.iitg.ernet.in/cse/ISEA/isea_PPT/ISEA_02_09/Presenataion_Flow_Base_Analysis.pdf

Netresec. (2014, May 1). *Network Miner*. Retrieved from http://www.netresec.com/?page=NetworkMiner

OASIS. (2013, December 12). *Electronic Identity Credential Trust Elevation Framework.* Retrieved from http://docs.oasis-open.org/trust-el/trust-el-framework/v1.0/trust-el-framework-v1.0.html

Popescu, A. (2012, May). *W3C Geolocation API Specification.* Retrieved from http://dev.w3.org/geo/api/spec-source.html

Courtney Imbert, courtneyimbert@gmail.com

Reddy, E. (2013, July 3). Neural Networks for Intrusion Detection and Its Applications. *Proceedings of the World Congress on Engineering 2013, Vol II.* Retrieved from http://www.iaeng.org/publication/WCE2013/WCE2013_pp1210-1214.pdf

Runnels, S. (2012, May 4). Learning the Bro Scripting Language: Practical Uses [Web log post]. Retrieved http://ryesecurity.blogspot.com/2012/05/learning-bro-scripting-language.html

Russell, S. & Norvig, P. (1995). *Artificial Intelligence: A Modern Approach.* Upper Saddle River, NJ: Prentice Hall.

Sarasamma, S., & Huff, J. (2005). *Anomaly-based techniques for Intrusion Detection Systems* [Presentation slides]. Retrieved from http://www.certconf.org/presentations/2005/files/RA2.pdf

Scarfone, K., & Mell, P. (2007, February 1). *Guide to Intrusion Detection and Prevention Systems (IDPS).* Retrieved from http://csrc.nist.gov/publications/nistpubs/800-94/SP800-94.pdf

Sen, S. (2007, March 19). *Performance Characterization & Improvement of Snort as an IDS.* Retrieved from http://www.tc.umn.edu/~ssen/papers/bell_labs_report_snort.pdf

Singh, V. (2012, May 21). iptables Rules to Limit Time & Quota-Based Access [Web log post]. Retrieved from http://varinderjhand.wordpress.com/2012/05/21/iptables-rules-to-limit-time-quota-based-acces/

Sommer, R. (2007, December 1). *The Bro Network Intrusion Detection System* [Presentation slides]. Retrieved from http://www.icir.org/robin/rwth/bro-intro.pdf

Tomes, T. & O'Connor, T.J. (2013, March 21). Hide and Seek, Post-Exploitation Style. *SchmooCon 2013.* Video retrieved from https://www.youtube.com/watch?v=Ys86goB5MQw

Trusteer. (2010, February 2). *Reused Login Credentials.* Retrieved from http://landing2.trusteer.com/sites/default/files/cross-logins-advisory.pdf

Courtney Imbert, courtneyimbert@gmail.com

Appendix A.

Open Source & Free Solutions for Network-based User Profiling

At the time of this writing, no single open-source security solution appears to provide a risk engine that combines multiple events into a threat level and acts on it. The simple profiling shown in this paper was demonstrated using a combination of free and open source tools:

tcpdump for capturing a representative sample of network traffic. tcpdump is available for download at http://www.tcpdump.org .

wireshark and **tshark** for some manual analysis of network traffic, including identifying a summary of protocols/ports, source and destination addresses, and service-specific credential fields. The **GeoIP** database, integrated with Wireshark, provided geolocation for IP addresses. Wireshark and tshark are available for download at http://www.wireshark.org/ . The GeoIP database and associated Wireshark plugins are available for download at http://dev.maxmind.com/geoip/ .

NetworkMiner to extract credentials from captured network traffic, though this can also be accomplished with tools like **dsniff** or **SniffPass**. NetworkMiner has other capabilities that are useful in network forensics work, like file extraction. At the time of this writing, NetworkMiner has a free client, with more features included in premium versions. NetworkMiner is available for download at http://www.netresec.com/?page=NetworkMiner .

p0f for operating system and browser fingerprinting. p0f is available for download at http://lcamtuf.coredump.cx/p0f3/ .

Snort for the signature-based intrusion analysis, including host-based rules and rules that check the content of packets for credentials. Snort is available for download at https://www.snort.org/ .

The Bro Network Security Monitor for flow-based network analysis. This tool is likely the closest possible to a risk engine as discussed in this paper, since an analyst can create detailed scripts that check iteratively through a packet against multiple criteria. Bro is available for download at http://www.bro.org/ .

Courtney Imbert, courtneyimbert@gmail.com

Appendix B.

Commercial Solutions for User Profiling & Risk Analysis

Scanning network traffic for credentials and checking for anomalies is complex. Several proprietary software providers have incorporated network traffic analysis in identity assurance products, or vice versa:

DSGateway, Delfigo Security

http://www.delfigosecurity.com

DSGateway offers a multi-factor identity solution that includes analysis of human behavior. It is an authentication platform that includes behavioral identity attributes like keystroke style, device (user agent, operating system, monitor resolution, browser type) and geospatial (timestamp, IP address, location, hostname, proxy IP). DSGateway evaluates each user transaction, assigns a "Confidence Factor", and transparently provides the appropriate level of system access.

Authentication Manager 8, RSA

http://www.emc.com/microsites/authentication-manager-8/index.htm

The latest release of the RSA SecurID platform, Authentication Manager 8, boasts "Risk-based Authentication". SecurID's most prominent service is providing more traditional multi-factor authentication with hardware to represent the possession factor. However, it also assigns risk levels to successful password-based authentication by tracking risk indicators, including a device profile and network traffic characteristics like IP address.

IdentityView, ArcSight

http://www8.hp.com/us/en/enterprise-software.html

IdentityView correlates data from multiple sources, including Identity Management systems and event logs, to monitor the actions of privileged users for anomalous behavior. It correlates data between addressing systems (DHCP, Kerberos, and any logs that retain IP addresses) to attribute activity to individual users. IdentityView ties multiple user accounts to a single identity and calculate a threat level based on the activities of the person associated with that identity (HP ArcSight, 2012).

Courtney Imbert, courtneyimbert@gmail.com