SANS Institute
InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

Detect, Investigate, Scrutinize and Contain with Rapid7 UserInsight

A review of Rapid7 UserInsight by SANS senior analyst Jerry Shenk. It discusses a tool that highlights user credential misuse while tracking endpoint system details that would be valuable to an incident response team.

Copyright SANS Institute
Author Retains Full Rights
Detect, Investigate, Scrutinize and Contain with Rapid7 UserInsight

A SANS Product Review
Written by Jerry Shenk
October 2014

Sponsored by
Rapid7

©2014 SANS™ Institute
Recent security breaches show that even companies with good defenses often don’t find out their systems are infected until outsiders tell them about it. Although the obvious goal for an IT security organization is to prevent a systems breach, we must assume our networks will be breached despite our best defenses. So, in addition to a good, flexible and updatable defense system, enterprises must have a response capability that detects a system compromise and takes action, such as sending alerts, sooner rather than later.

The problem is that “sooner” usually isn’t happening. According to the 2014 Verizon Data Breach Investigations report, half of all breaches remain undiscovered for months or longer. 1 Although logs may have collected evidence of these breaches as they happened, the difficulty of correlating that data with attacks quickly enough to enable an effective response to be mounted can be daunting even for the largest organization.

Collecting logs from different types of devices and merging them into useful alerts is the big challenge in log analysis. In the 2014 SANS Log Management Survey, more than one-third (36 percent) of respondents indicated that correlation or normalization of logs from disparate sources was their most challenging problem; two-thirds placed correlation among their top three challenges when integrating log collection and analysis with other security management tools. 2

To make matters worse, it’s not enough to monitor your internal systems. Security concerns are also reaching outside the local network, with detection getting harder to manage, largely because of the increasing adoption of cloud and mobile technologies. The need to monitor such services has been evident since 2011, when the SANS Log Management Survey first included questions about log collection from cloud and mobile sources. 3

---

Executive Summary (CONTINUED)

The leading cause of breaches—according to multiple reports—is user account compromise, whether through successful phishing, man-in-the-middle attacks, risky user behaviors or other means. Legitimate accounts are the Holy Grail for attackers who use what the network considers legitimate access to move laterally across systems and resources, send data out of the network and more. Even with the best protections, ask yourself:

- How do you detect compromised credentials?
- How certain are you that the cloud providers you rely on are doing a good job of securing your data?
- How certain are you that your employees haven’t leaked a password?
- Do you know how many login attempts have been made to your in-house and/or cloud resources? Do you know what those logins mean?
- Are you confident that all employees who have left the company have had their corporate cloud accounts disabled?
- How do you know all that movement on your network is legitimate traffic coming from legitimate user accounts?
- How secure are your critical internal systems from unauthorized login via internal accounts?
- How do you monitor administrator activity on your network?
- Can you detect attacker movements on your endpoints?

Rapid7’s UserInsight is designed to monitor user activity and detect compromised credentials across internal systems, outward-facing services (e.g., web or VPN access) and cloud providers, while identifying vulnerable assets and monitoring risky accounts. This is done by creating a baseline of known traffic patterns, geolocation and other user information, monitoring the local network, and keeping an eye on company accounts with cloud providers such as Google Apps, Amazon Web Services (AWS), Salesforce and Box. The goal is to identify attacks before a data compromise occurs and to sound an alert if a successful compromise is detected, while providing contextual information about users, devices and access behaviors so you can make informed decisions on how to act.

Overall, we found UserInsight to be useful for identifying compromised user accounts, providing alerts and enhancing visibility into the traffic and endpoint-related indicators of compromise. We found that most things worked very well. During our evaluation, Rapid7 slipstreamed several improvements to the collection and monitoring features of the product; such improvements are required of any security monitoring company that hopes to keep up with the constantly changing attack landscape.
UserInsight is a cloud-based network security monitoring service hosted by Rapid7 that uses server and site redundancy to ensure that no single failure can bring its service down. Organizations subscribing to UserInsight (its pricing is based on user count) place one or more collectors on their networks that report to Rapid7’s monitoring service. Organizations with simple networks can often get by with a single collector; organizations managing multiple network segments (e.g., branch locations) are most likely to use multiple collectors. The Rapid7 service also collects data from cloud service providers, aggregates it into a dashboard that shows current activity for the subscriber—both internally and in the cloud—and sends out alerts when an actual compromise has been detected. Figure 1 shows the dashboard from Rapid7’s demo environment.

![Figure 1. Typical UserInsight Dashboard](image-url)

*Identifying details have been redacted.*
A collector runs on a networked computer; deployments can feature multiple collectors, placed according to the organization’s size and geographic locations. The collector reaches out to internal servers to get Active Directory, LDAP, DHCP, DNS and other information from agentless endpoint monitors on Windows computers. It then passes this information to UserInsight servers in the cloud for analysis. The collector can also function as a syslog server to collect event data from firewalls and other devices that support syslog, such as our testbed’s firewall. The collector can also pull “flat file” logs such as those from DNS, DHCP and other servers and include them in the analysis.

It is also possible to scan endpoints to track users, local logins and local processes without having to deploy an endpoint agent. If unusual processes (such as known malware or processes that are unique to a single host) show up on networked devices, you may have an incident that should be looked into. UserInsight keeps track of information about users to identify deviations from normal activity that may indicate the user’s account has been compromised. Some of the tracked information includes login location, users’ critical assets and simultaneous attempts to access the network from multiple countries.

If UserInsight detects an administrative or service login coming from outside the organization, it can note that as an incident and send an alert to designated email addresses. Although it’s common in many organizations for administrators to log in from outside the network, the best practice is to log in under a less-privileged account and then elevate privileges only as much as needed to complete administrative tasks. These rule-based alerts are preconfigured in the monitoring service to generate alerts when there is a serious incident and avoid excessive alerting; the rules can be disabled in the portal.

Critical network assets—such as a file server with sensitive information, a workstation that processes PCI information or the workstation used by someone who deals with company secrets—can be tagged as a critical asset. Tagging raises the bar on what kinds of activity generate an alert.

Rapid7 also collects information about any cloud services the organization uses. Many cloud services—including Google Apps, AWS, NetSuite and Salesforce.com—are supported; new ones are being added as demand justifies. The on-premises collector connected to the customer network comes into play here; it pulls the relevant audit logs—rather than relying on an agent in the cloud—to address cases where rules on the provider’s end restrict such access to devices on the customer’s internal network.
To detect instances of malicious activity, Rapid7 also makes a “virtual honeypot” available with UserInsight. This honeypot, which was simple to deploy, monitors for suspect connection attempts, then reports those connection attempts to the monitoring service. If an IP address attempts to connect to various ports on the honeypot in a short period of time, security personnel should treat this as a suspicious event warranting further investigation, because the honeypot is not hosting any legitimate services and no internal computers should even know it exists.

Inversely, if there are legitimate systems that need to scan the network for legitimate reasons, they can be whitelisted in the honeypot’s alerting process and, therefore, won’t be flagged as malicious.

UserInsight also has a “honey user”; similar in concept to the honeypot, this user account could be referenced in internal documents, comments in web pages and other places that an attacker’s reconnaissance might find it. Nobody would “accidentally” try to log in as the honey user, so any attempted connection is suspect and suggests somebody is trying to gain illegitimate access.

Both the honeypot and the honey user can be deployed with as many instances as needed, depending on the physical and logical needs of the organization.

Two of UserInsight’s features—mobile device monitoring and threat ingress/egress monitoring—were beyond the scope of our review.
The testbed was a VMware environment with a Windows Server 2013 R2 domain controller also running DHCP and DNS services—with additional servers running Windows Server 2008 and a mix of clients running Windows 8.1, Windows 7 and Windows XP. The computers from our simulated “overseas” networks ran a mix of Windows and OS X. We set up several dummy user accounts in the Active Directory domain; some of these would log in to the mail server—Exchange 2013 running on a virtual server—through Outlook Web App (OWA).

Our installation of UserInsight started with a discussion with a Rapid7 engineer about the network to be monitored and what resources existed; following that, we downloaded and installed the UserInsight collector on a virtualized Windows 8.1 workstation. This workstation had to be able to establish an outbound connection to the Rapid7 cloud. The collector works best if it’s unhindered by proxies or traffic restrictions; outbound communication with the Rapid7 monitoring service (from both collector and honeynet) uses TCP port 443.

The collector installation was straightforward; once the collector was running, the rest of its configuration was done through a browser-accessed management portal. The test network’s firewall was already configured to allow outbound HTTPS, so no additional work was needed before we could proceed. We began the setup from a Windows client with the Aviator browser and also used Chrome and Firefox on Windows, as well as Safari on a Mac, without a problem. (Mozilla Firefox and Google Chrome are supported; Internet Explorer is not.)

Our first configuration task was to get the collector to synchronize with Active Directory; since it does this using LDAP, there really isn’t much to set up in current versions of Windows Server. This sync process is vital to UserInsight, because the directory holds the list of users being monitored for access and usage; this applies for cloud monitoring as well as the local network. We then set up UserInsight to monitor our Google Apps accounts so when an account was disabled in Active Directory, a login to Google Apps by that user would generate an incident.

Once the collector was synced to Active Directory, we set up collectors to pull the DNS and DHCP logs from their host. (The default directory path for each is buried in the Windows filesystem; therefore, on the recommendation of Rapid7 support, we changed that to directories right off the root, making it easier to find the logs and shortening the filesystem path.)

By default, UserInsight’s LDAP collector syncs once daily with the designated LDAP server. We changed this to an hourly sync, primarily because we were changing the testbed frequently. Although some environments may be fine with a daily LDAP sync, many should probably do this more frequently.

5 Although Microsoft no longer officially supports Windows XP, it still has a presence in many organizations, justifying its use in our testing.
The next thing on our checklist was a UserInsight virtual honeypot, which Rapid7 provides as a packaged appliance in Open Virtualization Archive (OVA) format, so installation was a matter of a few clicks to download the file and deploy it from the vSphere Client.

Once the in-house basics were running, we created a cloud services account on Google Apps for our testing. Often, cloud services aren’t directly connected to Active Directory; when a user is terminated, that must be reflected manually in the cloud services’ security profiles, but these are often missed. UserInsight can generate alerts when accounts that have been disabled in Active Directory are used at a supported cloud provider.

For organizations struggling to manage accounts locally and in the cloud, this notification is invaluable, but as anyone who has worked with log analysis knows, she is at the mercy of whatever generates the logs. After setting this up for Google Apps, we could receive reports of login successes and failures; by default UserInsight sends an alert after two failed connections to the configured account. (The output from Google Apps was a bit erratic, with alerts received hours after the event; other cloud vendors may have fuller logging.)

We also set up the agentless endpoint monitoring, which uses a collector that reaches out to the workstations to collect local information. It was a simple configuration in the management portal, where we:

- Named the collector
- Provided the name of the Active Directory domain
- Supplied the collector with local administrator credentials for the computers it was to monitor
- Selected an IP address range corresponding to the computers in our lab

Implementers should note that a collector can support multiple IP ranges and the Active Directory domain should be listed as its NetBIOS name, not the fully qualified domain name (FQDN).

With our testbed in place, we could start our evaluation of UserInsight in earnest: first by examining its ability to look for signs of account compromise, then by simulating attacks on user accounts.
Detecting Account Compromise

UserInsight is designed to detect malicious user actions, such as compromised credentials, across multiple applications and locations by detecting network patterns (lateral movement between devices), to scrutinize new samples of malicious access (through honeypots), and to contain activity by linking actions to known, acceptable user behaviors.

We wanted to start the review by examining whether it would detect suspicious use of credentials under the following scenarios:

- Using an account that has been disabled
- Using an account at two separate locations at the same time
- Using an account on an abnormal resource

Some of these issues, such as using a disabled account, should generate an immediate alert and, when detected, cause UserInsight to generate an alert. Logging in from an abnormal workstation does not, by itself, generate an alert but, in combination with other events or a critical asset, it can.

**Access from a Disabled Account**

Attempts to log in from a disabled account are always suspicious and can indicate an attack, but a failed login is not evidence of a successful attack. In contrast, when a user is disabled in Active Directory and successfully logs in to a monitored cloud service, that successful login should generate an alert. For this test, we disabled an account in Active Directory and then attempted to log in with that account from various simulated locations. An attempt to use the disabled account login from a workstation on the network failed and, as expected, did not trigger an alert. Another attempt to log in using OWA from a private IP address (10.1.1.170) failed and also did not trigger an alert. The third attempt to log in was from the IP address 14.0.75.51 (a simulated network using a South Korean address space); much to our confusion, this generated an alert even though the login was denied.

Now it was time to test this in the cloud, so we successfully logged in to Google Apps with the account that we had disabled in Active Directory. (Again, cloud accounts are often mistakenly left active, and thieves can use such “orphaned” accounts to get into the cloud service and find out information about the network, applications and data to target.)

Since our user account had not been deactivated on Google (as it should have been), our login was successful. However, when UserInsight pulled the Google Apps logs, it generated an alert because that account was disabled in Active Directory.

---

6 We researched this with Rapid7 support and found that the latest versions of OWA and the Microsoft IIS web server will log a failed OWA login attempt with the HTTP status code “200”; this typically denotes success, but is officially defined as “the request was fulfilled” — which it was, in a sense.
Access from Conflicting Locations

UserInsight also declares an incident when a single user logs in from multiple locations. The goal of this feature is to detect compromised user credentials because users should generally not be logging in twice at the same time. There’s a possible exception for two computers—but if those computers are in different geographic locations, such exceptions ought to be rare and can be defined as exceptions when they occur.

To test this, we set up a router in our lab with multiple IP address ranges that simulated IP addresses coming from Korea, Germany, New Jersey, Massachusetts and Los Angeles. We then set up computers in those IP address ranges and logged into Exchange using OWA. Initially, we connected as a user from our simulated “Korea” and “Germany”; within a few minutes we saw an alert on the screen and received an email telling us that a user had logged in from multiple countries, including the time between logins. (By default, alerts go to the email address used to log in to UserInsight.) The locations are highlighted on the map shown in Figure 2.

![Figure 2. Alert from Suspicious Login Incident](image-url)
Here, we can see that **test21 user** authenticated from these two countries 6:21 minutes apart. If the time were greater and these locations were a bit closer, we might need the detail to determine whether this was in fact an incident or not. By clicking on the red lightning bolt (indicated with a green arrow), we could see more detail of this incident. As shown in Figure 3, there clearly is cause for alarm—in the real world, our user couldn’t possibly have traveled from Korea to New Jersey in six minutes.

![Figure 3. Alerting to Login from Multiple Countries](image)

From this list, we can click on an incident and get the details about what caused it. We could also click on the username to see what other information about this user exists; if a machine name was associated with the incident, we could click on that as well.

Figure 4 shows the result of clicking on the incident **MULTIPLE COUNTRY AUTHENTICATIONS**: a map displaying the simulated geographic distribution of logins after a bit of testing, with records from “Massachusetts,” “Los Angeles,” “New Jersey” and “Korea.”
Detecting Account Compromise (CONTINUED)

Figure 4. Geographic Distribution of Logins
To further probe UserInsight’s ability to evaluate conflicting locations, we purposely failed several connections to the VPN from our simulated Korean and German networks, and then successfully connected to the VPN from both locations, with a four-minute delay between logins. Once the second login was successful, UserInsight generated an alert because there was a successful login from two separate locations. (It does not generate a report for a single failed login from two locations.) This is an important distinction; although it is certainly important to know that somebody is attempting to log in as a user on the network, the response should be very different if a login from a location where the user is known not to be present is successful, compared to the response to a failed attempt. Logging in from locations where the actual user can’t possibly be can indicate that the user has shared his credentials or that those credentials have been compromised and could be in use by a hostile entity.

**Harvested Credentials**

Another interesting attack scenario is one where the attacker attempts to connect with multiple, valid network credentials from a single location. The threshold is three failed attempts to log in with valid accounts. (Whether the access attempts succeed or fail, they are an area for concern that suggests valid user credentials may have been harvested.) To test this, we again purposefully failed to log in as three different users over a relatively short period of time from a single location, and these actions were detected and reported as an incident. We then ran failed logins twice with two valid user accounts and then a third time with a username that does not exist. No access was granted, and, because no access was granted and only two valid attempts were attempted, this was correctly not detected as an incident.

When we failed to log in as a third account, the rule was tripped and the incident was generated. We then successfully logged in as a fourth and separate user. This mimics a real-time scenario where an attacker would harvest credentials and then keep trying each one until a password is guessed—success after multiple failures—so it would be good to see an incident generated after a number of failed logins followed by a successful login.
Honeypot

Honeypots have a reputation for being notoriously complicated and difficult to set up. By contrast, the UserInsight honeypot couldn’t be simpler. Assuming you already have a working virtualization environment, you can just download the honeypot and start the installation with a couple of clicks, then answer the usual questions about IP addressing and let it run. Any number of honeypots can be deployed as needed to reflect the network topology.

The biggest decision we faced was what to name our honeypot and how to make it tempting to would-be attackers: you want something that looks legitimate and which might indicate a lower degree of machine security or the presence of forgotten but valuable data; disguising the honeypot as a development or archiving server can make the bait look even better. Development servers (e.g., SQLDEV04) are often not fully locked down; servers named BACKUP or ARCHIVE are probably not actively used but could have lots of valuable information. A good naming convention for an organization will make allowances for such “dummy” servers.

The UserInsight honeypot sits on the network waiting for connection attempts. If an attacker or some kind of malicious software starts scanning the network looking for open ports, the honeypot is there waiting to send a silent alarm when it gets a connection. A server that is not advertised and serves no purpose but gets a connection is probably a good indication that something abnormal is going on. When it receives a single connection attempt, it just waits, but a connection attempt to a second port—a common “doorknob-rattling” technique—is flagged as an incident and the Rapid7 monitoring service sends out an alert.

UserInsight also offers an expansion of the honeypot idea, the “honey user.” This is a user account in Active Directory that exists only as bait for a hostile attacker’s login attempts. The user isn’t connected to a real person; it’s just a trip wire account—that is, an account that should never have activity and when it does, it’s a good indication that somebody is up to no good.

Should an attacker touch either kind of honey trap, the system administrators can follow specific rules for response or start an investigation into the computer that triggered the alert.
To demonstrate the system’s detection capabilities, we created an attack scenario based on methods commonly used today, including:

1. Our attacker first harvests credentials from the corporate websites, email messages and news articles. From that research, a list of possible usernames is compiled, along with a list of common passwords to try.

2. We then tried logging into the organization’s VPN using these credentials. In this case, we tried five user accounts before we successfully connected as an authorized user.

3. We then ran a quick port scan of the connected network. We found a machine named “W2012R2”—a good guess that this is a server—with TCP port 3389 (Microsoft Terminal Services) open, so we attempted to connect over Remote Desktop Protocol (RDP).

4. After a few tries, we managed to connect to W2012R2 as “Administrator” with the highest level of system privilege. (In a real attack, a successful login this quickly might be a little optimistic, but it’s certainly possible, since a lot of admin accounts are left with default passwords or easy-to-guess passwords.)

5. An attacker wants to move quickly before detection, so once logged in as administrator, we created a new user with domain admin rights and logged off. At this point, this user becomes a valid admin-level user to the system, with all the access and controls afforded to admin on this and probably other systems.

6. We then logged back in, using the new account.
Using the default configuration, UserInsight generated four incidents and four email messages as these simulated events unfolded in the test network. The events this simulated attack generated are listed, newest to oldest, in Figure 5.

Figure 5. Recent Incidents Dashboard

These alerts followed the attack progress, including:

- **Harvested Credentials.** This makes sense—as the attacker, we tried five logins, one of which was not a valid username in Active Directory. This is the first incident that was generated before the attack succeeded.

- **Domain Admin Added.** Generated when we added **Test77 User** as an Active Directory domain administrator.

- **Account Privilege Escalated.** Generated when **Test77 User** was added to the “Domain Admins” group.

- **Ingress from Domain Admin.** We have four of these; each time we (as the attacker) connected to the VPN, we generated another occurrence.
Responding to Exploratory Access Attempts

Once an attacker has granted himself domain admin privileges, he will likely seek out other resources, access them and see what those lead to. As we continue the review, we (as the attacker) are going to use our seemingly legitimate access to roam and explore the network for a bit.

In our scenario, we (the attacker) mapped drives to additional computers and then started Terminal Services to use RDP to take control of those computers. We then mapped a drive to the workstation and created a batch file to stop the firewall. After that, we started Terminal Services and ran that batch file from Windows Task Scheduler. The commands involved appear in Figure 6.

```
net use T: \win81-test21\c$ 
notepad t:\rdpstart.bat
   netsh firewall set opmode disable
   reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f
   schtasks /Create /S Win81-Test21 /RU Test77 /RP Password1 /SC ONCE /ST 11:33 /TN RDP-Start /TR C:\rdpstart.bat
```

Figure 6. Sample Commands for Disabling Windows Firewall and Starting Terminal Services

We went through the same steps on the other hosts, substituting the host name as appropriate. Now that our attack was underway, it was time to return to the UserInsight dashboard to see what it had to tell us.
Detecting Violations and Breaches (CONTINUED)

The home tab on UserInsight showed six incidents. One of the incidents, “Ingress from Domain Admin,” showed 13 authentications from **Test77 User** (the compromised account). We were most interested in what this account had been doing, so we searched for it, as shown in Figure 7.

![Figure 7. Search by Specific User ID](image-url)
This search returned a collection of user information, activity, incidents, accounts, assets, locations and VPN connections related to this user, all in one place. (UserInsight’s search function supports a range of options, including searching by IP addresses, machine names, malicious URLs or listings in Rapid7’s Nexpose vulnerability management platform.) Most importantly, at the top, we can see the user was connecting as a domain admin, as shown in Figure 8.  

![Figure 8. Incidents by User](image)

At the bottom of the screen, we can see three incidents directly associated with **Test77 User**.

---

7 We already knew that, but in a real investigation, this is where antacids would come flying out of desks.
We clicked on the Assets link in the sidebar, and UserInsight presented a list of computers that **Test77 User** had accessed. What we saw here was a server (W2012R2) and two workstations that had been accessed by the attacker, as shown in Figure 9.

<table>
<thead>
<tr>
<th>HOSTNAME</th>
<th>LAST ACCESSED</th>
</tr>
</thead>
<tbody>
<tr>
<td>w2012r2.mytestlab.info</td>
<td>11:59:46 AM 8/16/14 (UTC-04:00)</td>
</tr>
<tr>
<td>win81-test21.mytestlab.info</td>
<td>11:35:59 AM 8/16/14 (UTC-04:00)</td>
</tr>
<tr>
<td>win81-test22.mytestlab.info</td>
<td>11:50:01 AM 8/16/14 (UTC-04:00)</td>
</tr>
</tbody>
</table>

**Figure 9. List of Assets Accessed by Suspicious User**

Clicking on the server’s entry, we saw a screen of detailed information from the server’s perspective. We were most interested in the other computers, but this screen contained useful context for our investigation, including related incidents, users, authentications, activity, IP addresses that the resource used, and processes running on the resource.

These all had interesting information, but we needed to know what other computers W2012R2 had connected to and, especially, whether **Test77 User** was involved. The visual displays of the attackers’ movements were also helpful. Scrolling through this information, we came to “Last 10 Authentications”; a connection map that shows how the attacker (“test77”) used the server as a “launch pad” to jump to two workstations, as depicted in Figure 10.

**Figure 10. Connection Map for Simulated Attack**
If this had been an actual attack and the user were left free to roam the network overnight, there would probably have been more than two servers taken over by morning, of course.

In a larger network configuration, simulated by Rapid7 and shown in Figure 11, we see a connection coming into the asset we are researching (green circle in the center) and from there connecting to nine other assets, one of which has been marked as “critical” (the red circle).8

![Figure 11. Anonymized Connection Map for Actual Authentication History](image)

When looking at just this view, we didn’t know whether this was an attacker or legitimate traffic, but we could see the relationship between assets, which is critical to incident response, repair and forensics.

The entire UserInsight application is designed around digging deeper for information and clicking on pieces of information for comparison with other data collected across the network from endpoints, firewalls and critical server endpoints (e.g., DNS, VPN or DHCP services). Working through this simple attack scenario gave us a good feeling for UserInsight’s capabilities. As with any monitoring tool, an organization should evaluate it in the production environment before full deployment.

8 Again, identifying details have been redacted.
Based on our experience, we found Rapid7 UserInsight to be helpful on a number of levels:

1. Detecting access violations under a variety of scenarios
2. Prompt alerting of only high-severity issues (minimal alarms)
3. Providing visibility and graphical elements to simplify response
4. Offering a safe means of detection through easily deployed honeypot technology that is part of the product
5. Scaling from small- to medium-sized business to enterprise use
6. Building in basic detection policies that follow into the cloud
7. Following an attack through the network to aid in cleanup and discovery of damage done

UserInsight is helpful in identifying misuse of network credentials. UserInsight also tracks internal processes, internal IP addresses of assets, connections from one computer to another, and other internal details that would be relevant to discovering what had been done with a compromised account and when and where it had been used.

No security tool is capable of doing it all, but UserInsight does fill a big blind spot in many organizations by prioritizing the discovery of user-credential misuse. UserInsight shows good promise of becoming a valuable part of a network’s security management portfolio.
Jerry Shenk currently serves as a senior analyst for the SANS Institute and is senior security analyst for Windstream Communications, working out of the company's Ephrata, Pennsylvania location. Since 1984, he has consulted with companies and financial and educational institutions on issues of network design, security, forensic analysis and penetration testing. His experience spans networks of all sizes, from small home-office systems to global networks. Along with some vendor-specific certifications, Jerry holds six Global Information Assurance Certifications (GIACs), all completed with honors: GIAC-Certified Intrusion Analyst (GCIA), GIAC-Certified Incident Handler (GCIH), GIAC-Certified Firewall Analyst (GCFW), GIAC Systems and Network Auditor (GSNA), GIAC Penetration Tester (GPEN) and GIAC-Certified Forensic Analyst (GCFA). Five of his certifications are Gold certifications. He also holds the CISSP certification.

SANS would like to thank this paper's sponsor:

RAPID7
# Upcoming SANS Training

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

<table>
<thead>
<tr>
<th>Event Name</th>
<th>Location</th>
<th>Dates</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>SANS Oslo 2015</td>
<td>Oslo, NO</td>
<td>Mar 23, 2015 - Mar 28, 2015</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS 2015</td>
<td>Orlando, FLUS</td>
<td>Apr 11, 2015 - Apr 18, 2015</td>
<td>Live Event</td>
</tr>
<tr>
<td>RSA Conference 2015</td>
<td>San Francisco, CAUS</td>
<td>Apr 19, 2015 - Apr 22, 2015</td>
<td>Live Event</td>
</tr>
<tr>
<td>Security Operations Center Summit &amp; Training</td>
<td>Washington, DCUS</td>
<td>Apr 24, 2015 - May 01, 2015</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS ICS London 2015</td>
<td>London, GB</td>
<td>Apr 27, 2015 - May 02, 2015</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS SEC401 London</td>
<td>London, GB</td>
<td>Apr 27, 2015 - May 02, 2015</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Bahrain 2015</td>
<td>Manama, BH</td>
<td>May 02, 2015 - May 07, 2015</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Security West 2015</td>
<td>San Diego, CAUS</td>
<td>May 03, 2015 - May 12, 2015</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Secure India 2015</td>
<td>Bangalore, IN</td>
<td>May 04, 2015 - May 16, 2015</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Secure Europe 2015</td>
<td>Amsterdam, NL</td>
<td>May 05, 2015 - May 25, 2015</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS/NH-ISAC Healthcare Cybersecurity Summit</td>
<td>Atlanta, GAUS</td>
<td>May 12, 2015 - May 19, 2015</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Pen Test Austin 2015</td>
<td>Austin, TXUS</td>
<td>May 18, 2015 - May 23, 2015</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Melbourne 2015</td>
<td>Melbourne, AU</td>
<td>May 18, 2015 - May 23, 2015</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS ICS Security Training - Houston</td>
<td>Houston, TXUS</td>
<td>Jun 01, 2015 - Jun 05, 2015</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS ICS410 Vienna in Association with IAEA</td>
<td>Vienna, AT</td>
<td>Jun 06, 2015 - Jun 10, 2015</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Dublin 2015</td>
<td>Dublin, IE</td>
<td>Jun 08, 2015 - Jun 13, 2015</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Houston 2015</td>
<td>Online TXUS</td>
<td>Mar 23, 2015 - Mar 28, 2015</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS OnDemand</td>
<td>Books &amp; MP3s OnlyUS</td>
<td>Anytime</td>
<td>Self Paced</td>
</tr>
</tbody>
</table>