Global Information Assurance Certification Paper

Karin Moon
GSEC Practical Assignment v1.3
May 8, 2002

If You Build It, They Will Come: Effective Honeypots

Honeypots are a contemporary and rather fresh inspiration to Internet security. In the past, the only means for an organization to learn about their vulnerabilities was after an attack had been launched. A honeypot, or a system of honeypots, allows us to gain knowledge of our vulnerabilities by connecting a decoy system to the network so that we can allow the blackhat community, those that illegally hack into computers, to break into the system. We are then able to watch and record their actions with the systems, and learn about their techniques before or while they are used on our real network environments. Honeypots can be used to attract internal and external malicious network intruders. An external system is used to watch external intruders, whereas an internal system watches for internal intruders, but it is possible that external intruders find their way into an internal honeypot.

In the beginning, honeypots were tools only sophisticated professionals in Internet security were implementing. Today, commercial tools are available and are being sold to Fortune 500 companies. This paper focuses on the use of honeypots in a production environment by providing a discussion of the following: a definition of honeypots, the different types of honeypots, the effectiveness of honeypots, the honeypot deployment and placement within an existing network, the required logging, and policies that should be implemented. Production honeypots must be efficient by successfully containing the intruder and logging their actions. Organizations that use honeypots as part of their security solution should add honeypot guidelines to their existing security policy.

Honeypot Definition

A honeypot is an information system that is placed in a network for the purpose of being attacked by hackers who are unaware they are being watched. Honeypots can appear to be single machines that contain information that has been falsified or honeypots can look like the network of a police headquarters, a school, a doctor's office, or whatever you think would attract probing eyes. By watching a honeypot being probed or attacked, an organization can learn and take actions to provide a more secure network environment. The introduction of the term "honeypot" originated from Clifford Stoll's May 1988 article *Stalking the Wily Hacker*, in which he discusses the information he discovered while watching a hacker attack his system. He compares their methods of tracking the attack to "catching flies with honey." Since Stoll's article, extensive research has been done on honeypots and the value provided by using these methods of deception to learn the techniques of the blackhat community.

Experts in Internet security are beginning to categorize honeypots by the intentions of the person that deploys a honeypot. Their intention can be for *research* or *production*. A production honeypot is used by an organization to prevent a compromise or detect an attack. The honeypot may be similar to an intrusion detection system, in which the honeypot would alert the organization that a user is probing their network without permission. The organization wants to minimize their risks by understanding the vulnerabilities and the threats that they face. A research honeypot is a resource tool that provides information about the techniques that attackers use to compromise systems. Recently, research honeypots have been coined a "honeynet." A honeynet is a network of honeypots. These systems may have different operating systems running a variety of applications. A research honeypot is measured by its effectiveness to gather data on threats. The intention is to gain information about the trend of network threats.

Risks are identified by the combination of vulnerabilities and the threat that these vulnerabilities will be exposed. For example, vulnerability in an internal network may be the ability for one employee to see into another employee's computer. If the employees are completely honest and would not probe the network, then there is a low threat. Unfortunately, not all employees are honest and therefore, there is a chance an organization's employees are a threat. The combination of the vulnerability and the threat creates risk. In this example, an internal honeynet would help an organization discover the unknown vulnerabilities and identify the employees that are a threat, thus mitigating or decrease their risks.

Honeypot Functionality

The only reason a honeypot exists is to lure malicious network intruders. The system can be a real operating system or a tool that intends to imitate the actions of an operating system. Real applications and data that appear to be real can be loaded on the honeypot, or a tool that emulates applications. The system does not provide any other value to a network. Therefore, any interaction with the honeypot should be regarded as unjustifiable intrusion. Monitoring the honeypot involves recording the actions of the hacker, which include probes, viewing and deleting files, and malicious code uploads. All of this can be audited along with the keystrokes the hacker uses to perform these actions.

Organizations are beginning to view honeypots as deterrence to, or as an early warning sign of, attacks occurring on the network. Recently, crackers, which are malicious hackers, broke into an airline corporation's network. The crackers stole sensitive information and posted it to a web site. If the airline had installed honeypots, it may have been possible to delay the attack and possibly even prevent it from happening. The crackers may have even posted false information that was provided to them on the honeypot.

Several factors can determine the successfulness of a honeypot. The placement of a honeypot will determine its visibility to an intruder. A highly visible honeypot will more likely be probed. A system that runs an operating system with known vulnerabilities or an application with known vulnerabilities will attract more "bees to the honey." When faced with a hardened system, meaning one that has little or no vulnerabilities, or a system with known vulnerabilities, the hacker will choose the latter. If you are trying to charm a prying intruder, such as an internal employee, you should name the system "Employee Data" or "Finance Department." The level of interaction the intruder can have with the system determines the length of his or her stay. If trespassers do not find appealing data, then they will not waste their time. They will not stay long if their ability to exploit the vulnerabilities is obstructed. On the other hand, if a production or research honeypot is compromised and used to launch an attack on another system, the owners of the honeypot can be held liable. This is the main disadvantage of a honeypot. Determining the level of interaction the intruder can have with they honeypot can become a balancing act. You want to learn from the attack, but you do not want your system to be able to compromise other systems in the network. An individual or organization that is considering implementing a honeypot should seriously evaluate the ramifications of a compromised honeypot.

Who should use a honeypot?

To determine who should use a honeypot, we should compare the cost incurred, the risks, and the potential value. The cost of maintaining a honeypot has two factors. The least expensive component is the actual hardware and software. Almost anyone can take an old computer with a small amount of memory and connect it to the Internet. Also, commercial honeypot products are available that simulate several machines while running on a single machine. The price of these commercial honeypot solutions can cost more than $25,000. Although, there are a few solutions that are provided free of charge. The second component of maintaining a honeypot is more expensive – time. For every minute a blackhat spends on a honeypot, at least an hour of data analysis must be done by the organization. Most attacks last more than thirty minutes. A single attack could take more than a full forty-hour week of work. Will the knowledge you gain from an attack be worth the time that you spend analyzing the data? The answer to that question will be different for each organization.

Honeynets

Due to the high interaction that is required by the honeypot administrator, research honeypots or honeynets require devoted time and energy. Honeynets typically allow the attacker to compromise a system and launch attacks. Thus, the inbound and outbound traffic must be watched and regulated. Honeynets cause high risks and high costs. The

information gathered by a honeynet has potentially high value. We can learn about the hacking tools, the techniques, and even analyze the trends in the hacker community. It is possible that we can then warn organizations of new vulnerabilities and suggest that developers create patches for their systems. Organizations that create Internet security software or systems that are used to connect to the Internet will find honeynets useful because the information they learn about attacks can be used to create more secure products. It would be ideal if everyone could create a honeynet and share the information, but the high risks and high costs suggest that only the security professionals with excess time and money implement honeynets or a research honeypot.

Production Honeypots

A production honeypot is best implemented in conjunction with several different security layers - intrusion detection systems, firewalls, and network segregation. The costs and risks associated with production honeypots are dependent on the level of interaction it can perform with an attacker. A honeypot solution that has low interaction capabilities is easy to find at a small cost and does not provide much of a risk because there is usually no system that can be compromised. Highly interactive honeypots come with high risks and high costs due to the ability for the system to become compromised and the amount of time and money spent on auditing the system.

The detection of an external attack has low value because there are few actions that can be taken to prevent future attacks. For example, hacker or audit tools that are used to ping a system not only spoof their source IP address, but the tool usually varies the spoofed source IP address. An organization may have a policy to create a rule on their IDS to drop all packets that are from the source IP address that has previously probed their production honeypot. This policy is not effective. The spoofed IP address may be an IP address of someone that truly needs access to the system.

Although you cannot predict the skill level of the attackers that will be toying with your honeypot**,** the traffic will probably be similar to the types of attacks that you see on your firewall or intrusion detection system. The common script kiddie will keep you busy watching the honeypot. A sophisticated hacker may spend some time looking around the honeypot. A hacker that spends too much time in a honeypot may become angry, if he or she realizes they have been wasting their time in a honeypot. The last think you want is an angry hacker that knows your IP address.

An organization that has production servers that are critical to its core competencies could implement a honeypot that imitated a production server. The honeypot would serve only to watch for attacks. Once an attack occurred, it can be removed from the network and used to determine how or what has been compromised. The organization may then be able to use that to determine how their production servers should be fixed. To protect

yourself from the risk of compromising the network, place a honeypot with low interaction capabilities externally. A honeypot is great if an organization needs to study the habits of hackers or if an organization is trying to lure hackers for another reason. But, studying hackers and their habits are not the objectives of most businesses.

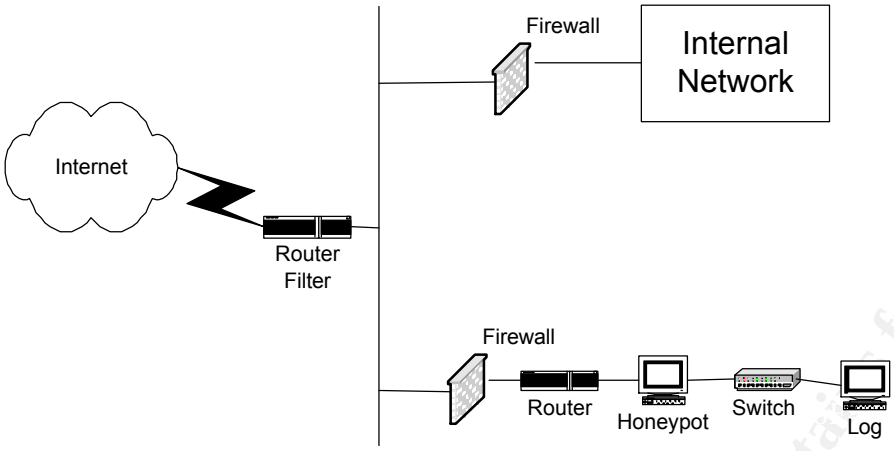
The greatest threat to a network is not the attack from the outside, but the attack from an internal employee. In today's corporate world, there has been a rise in angry laid-off employees and third party contractors. An employee already has access to your network and with a few probes can map the subnetworks. A production honeypot placed within the organization's network can provide high value because the organization will be able to identify the user and take appropriate action to prevent data and systems from being compromised**.**

Production Honeypot Placement

The placement of a honeypot in an internal and external network will determine its effectiveness. The network design should protect your network, but still allow data gathering and data control. You want the attacker to have enough freedom to be lured into your system, but not be able to use it to launch an attack in your environment or anyone else's environment. Layering your security is just as important in honeypots than it is in your real network. By adding layers that surround your honeypot, you are protecting your network from a compromised system. Honeypots should be used as tools for data gathering, but should not be used as your only data gathering mechanism. Data control is the key to an effective honeypot. You must be able to control whether an intruder has access to your honeypot or your real servers. Data control also includes managing the information that is leaving your network. Below are suggestions for the placement of an external honeypot and an internal honeypot.

External Honeypot Placement

To learn the most about your network's vulnerabilities, the placement and the filters you establish for the honeypot should reflect your real production network. If you have Linux running on your workstations and want to learn how it can be exploited, use the same version with the same patches on your honeypot. If you want to learn more about how a web server can be attacked, setup a system that closely imitates a "live" production web server in your organization. The placement of the honeypot allows you to control the data and access. The diagram below is one suggestion for a single workstation honeypot solution. This design may not be the best solution if you plan to have the honeypot emulate a web or mail server because the honeypot may then require different layers of security and access control.



If the honeypot is used to determine vulnerabilities of the internal systems, the firewall should have the same rules for the traffic that is destined for the honeypot and the traffic that is destined to the real internal network. If you filter traffic to your real production network, apply the same filters to your honeypot.

Rules should be established on the firewall that automatically triggers a page to your network administrator when traffic is directed to the honeypot. An email generated when the honeypot is being probed is not sufficient. What if the security professional is at home without access to email? Or the security professional could be eating lunch and not see the email until after the system has been compromised. If possible, have your firewall generate an email to you, and have any emails from your firewall consequentially send your mobile phone a text message. As stated before, no one should be intentionally accessing this system. So any traffic is suspect to investigation.

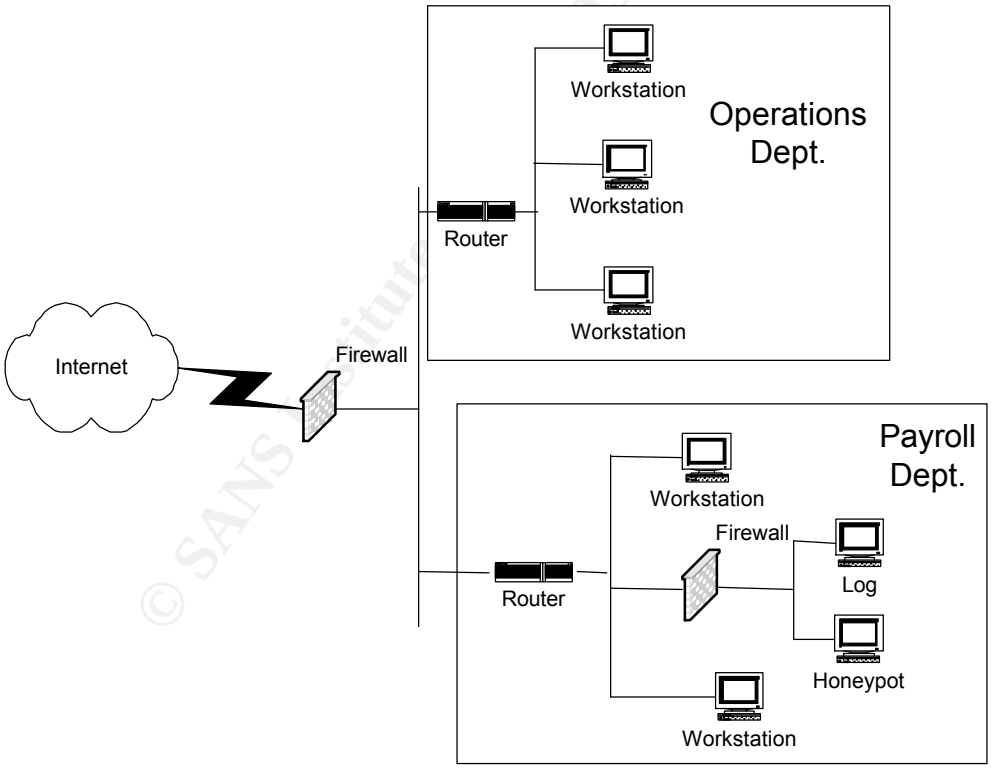
The router closest to the honeypot should not allow outbound packets to have a spoofed source IP address. If the firewall were to have this egress filter rule, the intruder in the honeypot would see the firewall. When firewalls are not transparent to intruders, they can attack its vulnerabilities by determining the vendor and the operating system. This router also serves as protection for the firewall. If an intruder is able to compromise a honeypot, the attacker will only be able to establish the devices touching it – in this case it would be a standard router with egress filtering.

One of the most important firewall rules in any honeypot system is to not allow the honeypot to be able to connect to your real network under any circumstances. The firewall rules must also monitor the types and the number of attempts of outbound traffic to the Internet. For example, allow the attacker a limited number of outbound ftp connections per day. This will keep you from being overwhelmed with the logs. It will also prevent the attack from getting out of hand. In the beginning, allow only a small number of outbound attempts on services with which you feel comfortable.

Overwhelming yourself with outbound traffic that you do not understand is best way to allow an attack to take over your network. On the other hand, remember that allowing outbound traffic is important for a successful honeypot. An attacker will not stay long if he or she cannot use the system to their advantage. The firewall should allow the ftp, http, udp, and stmp ports to send requests. The blackhat community use either ftp or http to download the necessary files from a server or a web site. Udp is necessary for domain name lookups. If an intruder compromises systems on a regular basis, he or she may keep track by sending the honeypot system information to themselves as an email – anonymous, of course. The firewall must allow outbound smtp from the honeypot.

Internal Honeypot Placement

An internal honeypot will alert you if an employee attempts to snoop around. The intent and method of an internal attack are different from an external attack. Employees have a greater fear of being caught, so their attacks or intrusions are sometimes harder to detect. The illustration below is an example of an internal network that contains several subnets and a single honeypot in the Payroll Department.



Within a network there may be several networks separated by routers. In this example, the honeypot is a database that can be accessed. The database would contain information that relates to the company, but would not provide any true information about employees or departments. This particular internal honeypot serves the purpose of catching intruders that intend to illegally access and change data. As with all honeypots, traffic to the honeypot should alarm you. Although it is possible that some innocent employees may accidentally attempt to access server, they will not pry further once they realize their mistake. The firewall should capture all data inbound and outbound to the honeypot. Have this firewall generate an alert via email. The email from your firewall should send a text message to your mobile phone.

The log server that is attached to the same firewall as the honeypot logs all of the traffic generated to the honeypot, but will not respond to any requests made by the honeypot. The ideal deployment of internal honeypots would require at least one honeypot for each internal department. For example, begin with the payroll department by placing a decoy server in the network. Financial organizations may consider placing a honeypot on each of their internal subnetworks that could be used to embezzle money or change account information. The possibilities are limitless and should be tailored to the risks that are associated with the organization's line of business.

Logging

Intruders are often clever enough to cover their tracks after they have compromised a system by changing the system logs. If they are really good, they will change the logs of all the visible systems. In order to learn from honeypots, we need to verify that the data we gather from the system is trusted. The best way to verify that a honeypot has data integrity is to log everything at every level and store those logs on an external device, away from the honeypot. The provided examples of an external and internal production honeypot do not allow for all of the following log suggestions. The crimes you are trying to catch with the honeypot will determine the placement of the system and the logging that is necessary.

Along with the alerts that are sent, the firewall should be logging all traffic to and from the honeypot**.** Load a network sniffer onto the same system that is running the firewall. The network sniffer should be able to record all inbound and outbound packets to the honeypot in their entirety. Although routers and switches have limited logging capabilities, they should be turned on. Tripwire, or software that is able to detect changes in files, is necessary to easily detect intrusion. Run this software on the honeypot itself. The honeypot should be logging all actions into the syslog files. Intruders, even the inexperienced have tools that replace the syslog files after the honeypot has been compromised. There are different ways to make the syslog record files locally as well as to an external device.

The illustrations present two methods - both are not perfect. The hacker will discover packets being sent to an external device if they have loaded a packet sniffer, like the one installed on the firewall. Log all the keystroke commands to the honeypot and external device, just as the other syslog files are. The information logged will be redundant, but if the honeypot becomes compromised you would hopefully be able to save at least some logs that have not been altered.

Honeypot Policies

It is important to have a honeypot security policy before any honeypots are actually deployed. Involve your legal department. The policy should include the actions that will be taken, if any, to prosecute external and internal intruders. Determine the actions that will be taken if an employee or contractor accesses a system and changes data. If internal honeypots are used, it would be prudent to include your honeypot policies as part of the employee policy handbook that requires signature. This will help to protect you and your organization from issues of entrapment. If the data gathered from a honeypot is going to be used for prosecution, discuss the possible privacy issues with your legal department. The information gathered, such as written code loaded onto a honeypot or typed conversations between two intruders on a compromised system, if shared with others for the purpose of learning, may be a violation of privacy if the attacker's authorization is not given.

An organization's security policy should already include the steps taken to prevent third party liabilities. In the policy describe the steps that have been taken to prevent the honeypot being used as a launching tool to attack others. This will protect the organization from being prosecuted for third party liabilities due to the compromise of a honeypot. Intruders are not always from the same country as the location of the attack. The network is truly global and legal issues should be addressed internationally as well as domestically.

Conclusion

The compromise of an organization's network is inevitable in today's Internet society. Ideally, we could all install honeypots, play head games with the intruders, and laugh at their confusion. This is not realistic. We must find a balance between the risk we face and the extent we will go to deceive the blackhat community. To make these decisions, consider all the factors. Deploying a honeypot cost money and valuable time. Do you have the expertise employed to monitor the honeypot? If not, what kind of risks are you acquiring? Are these honeypots going to provide any valuable information that you previously did not know about your network? These questions must be evaluated before you attempt to deceive the intruders.

Intruders are getting smarter by the minute and the only way to keep up with them are to watch their actions real time. In the future, production honeypots will be a considered part of a sound security solution. The solution to protecting your network is defense in depth. Honeypots are tools that provide another layer and another technique to block blackhat community threats. If you have the resources to add honeypots to your layers of defense and are willing to incur the risks, use the production honeypot as a tool. You will learn about hacker techniques and the vulnerabilities of your current network.

References

1. Lamb, Andrew. "Incident Analysis of a Compromised NT Honeypot." December 2001. URL: http://www.lucidic.net/whitepapers/alamb-12-2001.html (April 2002).
2. Lemos, Robert. "How much does it cost to track a hack?" *ZDNet News*. March 22, 2001. URL: http://www.customdigitalsolutions.com/hacktrack.htm (April 2002).
3. Holcroft, Stephen. "Design of a Default Redhat Server 6.2 Honeypot." April 2002. URL: http://www.lucidic.net/whitepapers/sholcroft-4-2002.html (April 2002).
4. Recourse Technologies, Inc. Honeypot Effectiveness Study. September 22, 2000. URL: http://www.recourse.com/download/press/releases/Honeypot.pdf (April 2002).
5. Spitzner, Lance. "Honeypots: Definitions and Value of Honeypots." March 8, 2002. URL: http://www.enteract.com/~lspitz/honeypot.html (April 2002).
6. Spitzner, Lance. "Know Your Enemy: Honeynets." January 14, 2002. URL: http://www.project.honeynet.org/papers/honeynet (April 2002).
7. Stoll, Clifford. "Stalking the Wily Hacker." May 1988. URL: http://cne.gmu.edu/modules/acmpkp/security/texts/HACKER.PDF (April 2002).
8. The Honeynet Project. Know Your Enemy. Boston: Addison-Wesley, October 2001.

Last Updated: July 20th, 2017

Upcoming Training

SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
Community SANS Omaha SEC401*	Omaha, NE	Aug 14, 2017 - Aug 19, 2017	Community SANS
Community SANS Trenton SEC401	Trenton, NJ	Aug 21, 2017 - Aug 26, 2017	Community SANS
Virginia Beach 2017 - SEC401: Security Essentials Bootcamp Style	Virginia Beach, VA	Aug 21, 2017 - Aug 26, 2017	vLive
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Chicago 2017	Chicago, IL	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
Community SANS Pasadena SEC401 @ NASA	Pasadena, CA	Aug 23, 2017 - Aug 30, 2017	Community SANS
Mentor Session - SEC401	Minneapolis, MN	Aug 29, 2017 - Oct 10, 2017	Mentor
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
Mentor Session - SEC401	Edmonton, AB	Sep 06, 2017 - Oct 18, 2017	Mentor
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
Mentor Session - SEC401	Ventura, CA	Sep 11, 2017 - Oct 12, 2017	Mentor
Community SANS Albany SEC401	Albany, NY	Sep 11, 2017 - Sep 16, 2017	Community SANS
Community SANS Dallas SEC401	Dallas, TX	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Columbia SEC401	Columbia, MD	Sep 18, 2017 - Sep 23, 2017	Community SANS
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Copenhagen 2017	Copenhagen, Denmark	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Boise SEC401	Boise, ID	Sep 25, 2017 - Sep 30, 2017	Community SANS
Baltimore Fall 2017 - SEC401: Security Essentials Bootcamp Style	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
Community SANS New York SEC401	New York, NY	Sep 25, 2017 - Sep 30, 2017	Community SANS
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Sacramento SEC401	Sacramento, CA	Oct 02, 2017 - Oct 07, 2017	Community SANS
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Oct 02, 2017 - Oct 07, 2017	Community SANS