Development of an Educational Data Acquisition System to Profile Cyber Attacks

Philip J Lunsford II, Erol Ozan, Lee Toderick, Tijjani Mohammed
East Carolina University
lunsfordp@ecu.edu
ozang@ecu.edu
toderickl@ecu.edu
mohammedt@ecu.edu

Abstract

A prototype course model is presented that is based on a live network-based honeypot to monitor network attacks. In this system a server supporting specific services that include SSH, HTTP, SMTP, and FTP is configured and set up behind a logging firewall. Advanced logging and reporting functions include login attempts, IP addresses, dates, times, and frequency of attempts. Students use the log files and employ filtering and data pattern analysis tools to analyze and profile the cyber attacks. The developed system constitutes a flexible data gathering platform that facilitates the classroom observations and experiments in the area of information security.

Introduction

Cyber attacks continue to become more sophisticated each day. Most attackers employ automated tools to penetrate the systems connected to the Internet. An important part of information security education involves the understanding of the dynamics of cyber attacks. This understanding can best be achieved by observing the real attacks as reported by Romney *et al* [1]. To this end, of particular interest is the use of honeypot technologies [2] to observe, classify, and defend against attacks. A honeypot or a honeynet is a real or simulated network or network resource that acts as a trap for attacks. Honeypots are used to investigate and analyze network attacks without compromising the production systems. Exposing students to live honeypots not only introduces them to this concept, but gives them a real world scenario with attacks that are not simulated. Permitting attacks from the Internet connection demonstrates to the students that network attacks are active in the real world, but also introduces some risks into the laboratory environment. Care must be taken to prohibit hackers from using any compromised honeypot machine to launch attacks.

Description of the Honeynet-based Live Laboratory Exercise

A learning module was added to the senior level network troubleshooting course at East Carolina University to provide student with exposure to current network attacks using a honeypot system. This module is based on the network topolgy shown in figure 1.

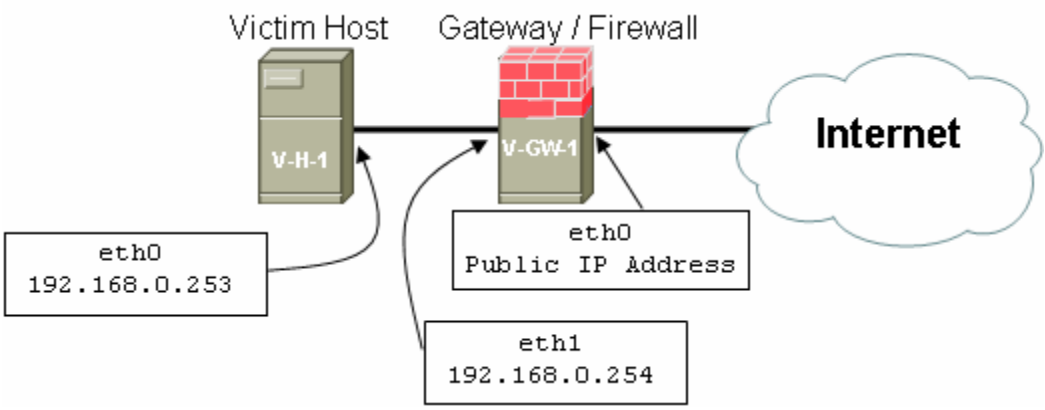

Figure 1. Network topology.

The "Top 10 Target Ports" from www.dshield.org/topports.php were evaluated for suitability for the honeypot, and either included in V-GW-1 logging or permitted through to the V-H-1 computer.

Host V-H-1 is a Linux Red Hat host, configured for HTTP, SMTP, SSH, and FTP services. To allow for analysis of changes to the system, this host has Tripwire [3] installed. Tripwire is a commercial product that easily allows a system administrator to audit changes. It acts as an independent control to detect, reconcile and report change. Thus V-H-1 is the baited system that is inviting attack.

Gateway V-GW-1 is a dual-homed Linux Red Hat host, configured for minimal services. The purpose of V-GW-1 is to act as a transparent gateway that will monitor traffic to the target host V-H-1 and also block any malicious traffic originating from V-H-1 when it becomes compromised. To allow remote access on V-GW-1, SSH server is running, but this machine is not intended as a victim machine. To help hide this SSH service to V-GW-1 but also allow SSH attacks on V-H-1, we allowed SSH traffic on the well known SSH port 22 [4] to pass through the firewall and be forwarded to V-H-1. SSH traffic on an alternate non-standard port number is allowed to connect directly to V-GW-1. Using a non-standard port number for this SSH traffic helps to reduce the risk of V-GW-1 being compromised, as ssh brute-force dictionary attacks overwhelmingly scan for ssh services on well-known port 22. This remote access was needed for the students to review and evaluate log files, as well as change the firewall rules. Linux netfilter is enabled on V-GW-1 allowing firewall rules to be applied using a script with iptables [5] commands.

To aid in logging, Network Time Protocol (NTP) was used. V-GW-1 received stratum 2 time from Internet-based clocks. V-GW-1 acted as the time server to V-H-1.

Configuration the iptables rules must be carefully considered. The V-GW-1 system must ensure that no attacks can originate from this system (either V-GW-1 or V-H-1), but attacks to the victim V-H-1 should be allowed. Attacking traffic with a destination address of the external eth0 interface of V-GW-1 should be passed on to the victim machine V-H-1. Traffic back to the attacker coming from V-H-1 should be allowed to pass through V-GW-1 so that the attack on the

victim machine is allowed to continue, but any traffic destined for a different external machine, potentially a new victim machine, must be blocked. Additionally logging of traffic in both directions must be maintained so that students may have data to analyze. For this network we chose to use the private address space of 192.168.0.0/24 between the gateway and victim machines. This necessitates the use of network address translation (NAT) [6]. Thus the firewall rules can be summarized as follows:

1. Drop all traffic not explicitly allowed.
2. Allow SSH traffic on the assigned alternate port, 743, to connect to V-GW-1.
3. Allow and provide NAT translation and forwarding to V-H-1 for the following protocols on well-known ports inbound to interface eth0 on V-GW-1: HTTP, SMTP, SSH, and FTP.
4. Allow and provide NAT translation and forwarding for traffic from V-H-1 to the original attacking machine on established connections.

Figure 2 depicts packet flow through V-GW-1 for HTTP, SSH, FTP, and SMTP traffic. When packets for these protocols are received by V-GW-1, the destination address is changed to V-H-1, then forwarded. Return traffic is passed through V-GW-1 to the attacker. Following are IPTABLES rules to accomplish this for SSH, along with the forwarding rule to permit return traffic. Variables $EXTADD, $EXTIF, and $VICHOST refer to IP addresses.

```
# REDIRECT SSH on port 22 to internal victim
  iptables -t nat -A PREROUTING -d $EXTADD -i $EXTIF -p tcp -m tcp --dport 22 -j DNAT --to-
destination $VICHOST
  iptables -A FORWARD -d $VICHOST -i $EXTIF -o $INTIF -p tcp --dport 22 -m state --state NEW  -j
ssh-connections
# PERMIT RELATED and ESTABLISHED traffic to pass
  iptables -A FORWARD  -p tcp  -m state --state RELATED,ESTABLISHED -j ACCEPT
```

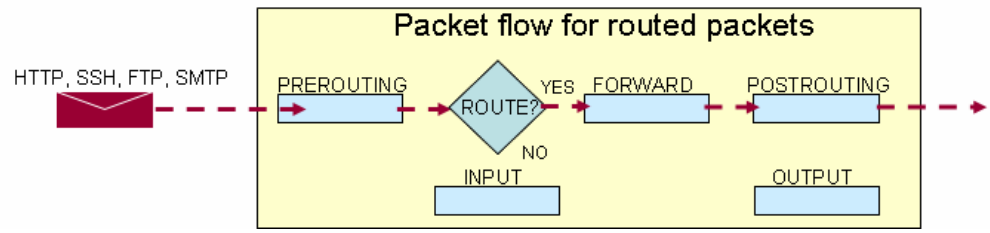

Figure 2. V-GW-1 routing for V-H-1.

Figure 3 depicts packet flow for all other traffic. Only TCP port 743 is permitted into V-GW-1 from the outside. UDP port 123, NTP, is permitted Following are IPTABLES rules that set this condition:

```
# allow ssh connections via port 743
iptables -A INPUT -p tcp –i $EXTIF -m tcp --dport 743 -m state --state NEW -j ssh-connections
iptables -A INPUT -p tcp -m tcp --dport 743 -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp -m tcp --sport 743 -j ACCEPT
# ALLOW NTP traffic
iptables -I INPUT -p udp --dport 123 -j ACCEPT
iptables -I OUTPUT -p udp --sport 123 -j ACCEPT
```

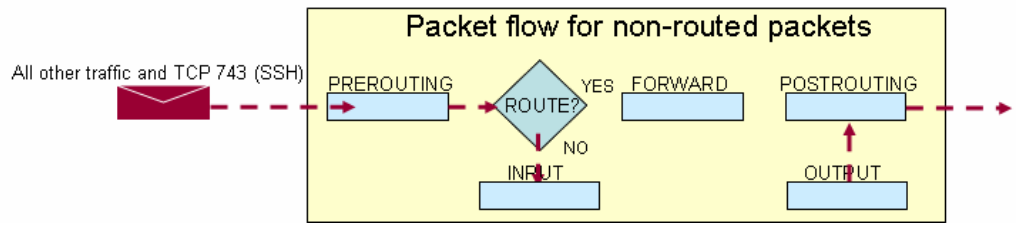

Figure 3. V-GW-1 flow for non-routed packets.

Gateway V-GW-1 also logs other UDP and TCP connections before blocking. For example, Dshield.org continually identifies UDP port 1026, Windows Messenger Spam., as a highly probed port. Likewise, UDP ports 1027 and 1028 perform similar functions. Customized firewall chains that log, then drop, the packets follow:

```
# log ALL NEW messenger spam (Windows RPC) packets.
iptables -A INPUT -i $EXTIF -p udp -m udp --dport 1026:1028 -m state --state NEW -j msngrspam-
connections
iptables -A msngrspam-connections -j LOG --log-prefix "IPTABLES-MSNGRSPAM-Traffic : "
iptables -A msngrspam-connections -j DROP
```

A partial transcript of the above rules actively filtering traffic can easily be gleaned by the student:

```
Chain msngrspam-connections (1 references)
 pkts bytes target     prot opt in     out     source               destination
   39 19216 LOG        all  --  any    any     anywhere             anywhere
LOG level warning prefix `IPTABLES-MSNGRSPAM-Traffic : '
   39 19216 DROP       all  --  any    any     anywhere             anywhere
```

Students evaluated /var/log/messages for all instances of “IPTABLES-MSNGRSPAM-Traffic” for probes to these ports. Students are warned, however, that UDP packets are subject to source IP address spoofing; that is, the source IP address of the packet probably did not originate from that computer. Following is a typical line from the log file, where DPT indicates a probe on port 1026:

```
Apr 15 21:36:15 V-GW-2 kernel: IPTABLES-MSNGRSPAM-Traffic : IN=eth1 OUT=
MAC=00:02:b3:28:87:3a:00:b0:c2:3b:10:00:08:00 SRC=204.16.209.20 DST=150.216.57.32
LEN=388 TOS=0x00 PREC=0x00 TTL=49 ID=0 DF PROTO=UDP SPT=36869 DPT=1026 LEN=368
```

SSH brute-force dictionary attacks were logged at V-H-1. Students saw first-hand that most of these attacks are prefaced by an initial scan of the network. SSH servers are discovered, then attacked. A partial transcript of this type of attack shows the initial scan, followed by a particularly lengthily attack of 1,670 separate login attempts:

```
23:26:26 V-H-2 sshd[27479]: Did not receive identification string from
::ffff:59.106.23.222
23:36:59 V-H-2 sshd[27972]: Invalid user setup from ::ffff:59.106.23.222
<output omitted>
00:16:22 V-H-2 sshd[30274]: Failed password for invalid user sharon from
::ffff:59.106.23.222 port 54131 ssh2
00:16:28 V-H-2 sshd[30277]: Failed password for invalid user vincent from
::ffff:59.106.23.22
```

For the spring semester of 2006, students were presented with the network shown in Figure 1 along with documentation. All students reviewed the iptables configuration script to understand

the function and operation of the gateway. The honeypot system was allowed to collect attacks and probes for 18 days. After this time students prepared a report that included:

1. A list of attacks by protocol and source including:
   a. Source IP address
   b. Country of origin.
   c. Count of the number of connection attempts for SSH attacks
   d. The purpose of the attack
2. A list of files and directories that were compromised
3. A list of suitable messages for each service that warns of potential legal action.
4. Suggested specific improvements to the firewall or other devices.
5. Suggested specific improvements to the laboratory exercise to make it a better learning experience.

The students identified 12 separate attacks from 10 different countries to SSH, TCP port 22. During this time, no attacks to SSH on V-GW-1 were logged. From the V-H-1 HTTP log file directory entries the students were able to conclude that at least one attacker may have been trying to compromise the victim machine to host an online forum. Note that the internet access from the campus network is protected by a university-maintained firewall. Configuring this same laboratory exercise outside of this firewall would have potentially allowed more attacks to be logged.

Assessment

Six of the eight students responded to a short four question survey to assess the student's perspective of the usefulness of the exercise. The results are shown in Table 1.

Table 1. Student Survey Results

Survey Question	**Strongly Agree**	**Agree**	**Disagree**	**Strongly Disagree**
The case study helped me understand the profile of real world attacks	4	2	0	0
The case study helped me understand how to categorize network attacks	2	4	0	0
The case study helped me understand how to develop a methodology for monitoring attacks	2	4	0	0
The case study helped me understand critical fields in log transcript entries	4	2	0	0

The informal feedback from students was also very positive. Analyzing actual current attacks gave the students a sense of urgency in learning and also added excitement to the exercise.

The work reported by Romney *et al* [1] used the tools available from The Honeynet Project [7]. Compared to our network, this network is more automated and provides a higher sophistication

and more victim opportunities as multiple victim operating systems were used. Our exercise, on the other hand, provided a more generalized framework that can be customized easily and also required the students analyze the firewall rules and to write scripts to analyze the raw log data. This required the students to gain an in-depth understanding of the log data, providing a deeper, but narrower analysis of the attacks.

Conclusion

Honeypot systems can provide an important roll in teaching security of information technologies. Analyzing current attacks on systems allows students to study up to date attacks and to increase awareness of threats to information assurance. Live laboratory networks can be set up with minimal cost and provide a valuable learning opportunity for information technology students.

Anecdotal results of this system are very exciting to both learning and behavior change. Several students firewalled their home computers, evaluated firewall logs, and started 'bragging-rights' for logs with the highest number of dropped attacks. Other students used 'Google Earth' to view cities and countries where attacks originated.

Because of the success with this prototype, the model will be expanded and modified for courses that cover in-depth defense and analysis of active attacks. In addition to attack concepts and theory, students are presented with live situations where a compromise could have negative consequences on the organization network. Future models will expand on firewall construction and filtering, packet mangling, attack signatures, and automated responses.

Bibliography

[1] Romney, Gordon W., Jones, Jeremiah K., Rogers, Brandon, L. and MacCabe, Philip. "ITSecurity Education is Enhanced by Analyzing Honeynet Data," ITHET 6th Annual International Conference, July 7-9, 2005, Juan Dolio, Dominican Republic.

[2] Thigpen, Seth D., Lunsford, Philip J. "Current honeypot Technology and Available Tools", under review. Submitted to The International Journal of Modern Engineering, December 2005.

[3] World-Wide Web URL http://www.tripwire.com/. Last Accessed June 14, 2006.

[4] World-Wide Web URL http://www.iana.org/assignments/port-numbers. Last Accessed June 14, 2006.

[5] Netfilter, World-Wide Web URL http://netfilter.org/. Last Accessed June 14, 2006.

[6] Egevang, K. and Francis, P. RFC 1631.World-Wide Web URL http://www.ietf.org/rfc/rfc1631.txt Last Accessed June 14, 2006.

[7] The Honeynet Project, World-Wide Web URL http://www.honeynet.org/ Last Accessed June 14, 2006.

Biographies

PHILIP J. LUNSFORD II received a B.S. in Electrical Engineering and a M.S. in Electrical Engineering from Georgia Institute of Technology and a Ph.D. in Electrical Engineering from North Carolina State University. He is a registered professional engineer and is currently an Assistant Professor at East Carolina University. His research interests include network security, telemedicine applications, and system simulation.

EROL OZAN received a B.S. in Electrical and Electronic Engineering from Middle East Technical University, a MS in Applied Physics from Istanbul University, and a Ph.D. in Engineering Management from Old Dominion University. He is currently an Assistant Professor at East Carolina University. His research interests include e-commerce security, web application development, and computer visualization.

LEE TODERICK received a B.S. in Computer Science from East Carolina University and an MS in Computer Information Systems from Boston University. His current professional certifications include CCNP/CCDP and RHCE. He is a Lecturer in the Department of Technology Systems at East Carolina University. Research interests include developing remote access courses for distance learning and network security.

TIJJANI MOHAMMED is a faculty member in the Department of Technology Systems, within the College of Technology and Computer Science at East Carolina University. Dr. Mohammed teaches graduate and undergraduate courses in Information Technology and has research interests in distance delivery of lab-based courses, secure remote access labs, web services management, and microprocessor applications.