# Global Information Assurance Certification Paper

*** Northcutt, Brent has probably learned a lot with his experience. Accuracy is fine overall, but then if you were creating the traces, shame on you if you don't know what they are! 74 *

# Intrusion Detection Practical

## 10 Detects of Intrusion

Brent A. Williams

04/20/00

**Intrusion Detection Practical**
This practical will demonstrate ability to detect various intrusion attempts. Some of the detects were produced in a lab, some were taken from GIAC, and some were actually caught in the wild. I have labeled them appropriately. For the ones caught in the wild, we have a packet filtering router and a firewall in place. For lab scenarios, neither of these devices were in place.

**Detect #1 (taken from GIAC)**

```
22:01:32.843383 209.73.241.116.46507 > XXX.XXX.50.2.8080: S 3746639756:3746639756(0) win 8760  (DF) (ttl 246, id 38853)
22:01:32.846907 209.73.241.117.46508 > XXX.XXX.50.3.8080: S 3746729679:3746729679(0) win 8760  (DF) (ttl 246, id 38863)
22:01:32.868900 209.73.241.104.46515 > XXX.XXX.50.10.8080:S 3747221799:3747221799(0) win 8760  (DF) (ttl 246, id 38883)
22:01:32.875482 209.73.241.105.46516 > XXX.XXX.50.11.8080:S 3747318196:3747318196(0) win 8760  (DF) (ttl 246, id 38893)
22:01:32.887872 209.73.241.106.46517 > XXX.XXX.50.12.8080:S 3747331897:3747331897(0) win 8760  (DF) (ttl 246, id 38903)
22:01:32.902460 209.73.241.109.46520 > XXX.XXX.50.15.8080:S 3747452632:3747452632(0) win 8760  (DF) (ttl 246, id 38913)
22:01:32.906017 209.73.241.110.46521 > XXX.XXX.50.16.8080:S 3747481685:3747481685(0) win 8760  (DF) (ttl 246, id 38913)
22:01:32.995200 209.73.241.102.46533 > XXX.XXX.50.28.8080:S 3748332241:3748332241(0) win 8760  (DF) (ttl 246, id 39013)
22:01:33.003663 209.73.241.104.46535 > XXX.XXX.50.30.8080:S 3748458349:3748458349(0) win 8760  (DF) (ttl 246, id 39023)
22:01:33.138686 209.73.241.118.46569 > XXX.XXX.50.2.8080: S 3750593619:3750593619(0) win 8760  (DF) (ttl 246, id 38854)
22:01:33.139445 209.73.241.118.46569 > XXX.XXX.50.2.8080: R 3750593620:3750593620(0) win 8760 (DF) (ttl 246, id 38855)
22:01:33.140214 209.73.241.119.46570 > XXX.XXX.50.3.8080: S 3750689204:3750689204(0) win 8760  (DF) (ttl 246, id 38864)
22:01:33.140990 209.73.241.119.46570 > XXX.XXX.50.3.8080: R 3750689205:3750689205(0) win 8760 (DF) (ttl 246, id 38865)
```

**Synopsis of Detect**
The above capture looks like a host scan for any host listening on port 8080. It looks as if these are crafted packets, most likely coming from the same machine. The scans are transmitted within a very close time frame, leading me to believe this was a script. This also seems to be a semi-sporadic scan, from looking at the pattern of hosts scanned (.2, .3, .10, .11, etc). The ttl field is also the same, and the fragment ids are very close in number.

| **Issue** | **Score** | **Reason** |
|---|---|---|
| Criticality | 3 | Depending on responses, may have a good network map |
| Lethality | 4 | May be lethal to web services or proxy if server responds |
| System Countermeasures | 5 | Assuming the OS is up to date and all patches are installed |
| Network Countermeasures | 4 | Assuming that any accessible machines will have this port disabled, the firewall would also block port 8080 |
| **Total** | **-2** | |

**Detect #2 (produced in lab)**
22:17:01.732231 netmap.attacker.com > 10.0.1.0: icmp: echo request

22:17:01.732234 netmap.attacker.com > 10.0.1.255: icmp: echo request

22:17:01.732241 netmap.attacker.com > 10.0.2.0: icmp: echo request
22:17:01.732243 netmap.attacker.com > 10.0.2.255: icmp: echo request
22:17:01.732249 netmap.attacker.com > 10.0.3.0: icmp: echo request
etc……

**Synopsis of Detect**
This is the most basic and obvious attempt at network mapping. The attacker may have been a little more sneaky and used a slow and low approach as opposed to flooding our network with ICMP requests on the network and broadcast addresses.

| **Issue** | **Score** | **Reason** |
|---|---|---|
| Criticality | 3 | Depending on responses, may have a good network map |
| Lethality | 1 | Attacker has useful network map for reconnaissance |
| System Countermeasures | 4 | Firewall is running very secure OS w/ updated patches |
| Network Countermeasures | 5 | Router eats icmp packets from external networks |
| **Total** | **-5** | |

**Detect #3 (produced in lab)**
19:03:11.312371 netmap.attacker.com > 10.1.10.2.25

19:03:11.312273 netmap.attacker.com > 10.1.10.2.23

19:03:11.312281 netmap.attacker.com > 10.1.10.2.53
19:03:11.312284 netmap.attacker.com > 10.1.10.2.21
19:03:11.312289 netmap.attacker.com > 10.1.10.2.69
etc……

**Synopsis of Detect**
This is an obvious port scan (of well known ports) on a host that may have replied to a host scan. It was probably an automated attack because of the time frame of the hits. The attacker may have gained the ip address information from some previous reconnaissance. This machine does not have access to any internal network machines. The firewall would deny any attempts of intrusion to our internal net.

| **Issue** | **Score** | **Reason** |
|---|---|---|
| Criticality | 4 | This machine is a UNIX system that was luckily locked down pretty tight |
| Lethality | 4 | A compromise on this machine isn't very threatening due to the fact we don't run NIS |
| System Countermeasures | 4 | Firewall is running very secure OS w/ updated patches |
| Network Countermeasures | 3 | Router eats icmp packets from external networks |
| **Total** | **1** | |

**Detect #4 (produced in lab)**

02:36:56.713850 spoof.attacker.com:117 > mynet.12375: S 8123593:8123593(0) ack 682235132 win 8192 <mss 1460>
02:36:56.713850 spoof.attacker.com:117 > mynet.6325: S 8123596:8123596 (0) ack 26158 win 8192 <mss 1460>
02:36:56.713853 spoof.attacker.com:117 > mynet.9025: S 8123599:8123599 (0) ack 2209571 win 8192 <mss 1460>
02:36:56.713853 spoof.attacker.com:117 > mynet.12576: S 8123602:8123602 (0) ack 14270 win 8192 <mss 1460>
02:36:56.713854 spoof.attacker.com:117 > mynet.1225: S 8123610:8123610 (0) ack 583 win 8192<mss 1460>
02:36:56.713856 spoof.attacker.com:117 > mynet.5443: S 8123614:8123614 (0) ack 462014 win 8192 <mss 1460>

**Synopsis of Detect**

This is an example of a spoofed address. There was no initial SYN, so this leads me to believe that our address may have been spoofed. This could also potentially be a DOS attempt. It looks as though it was a script, because of the short time frame all the packets were sent.

| **Issue** | **Score** | **Reason** |
|---|---|---|
| Criticality | 4 | This is a Windows machine that was luckily locked down pretty tight |
| Lethality | 4 | A compromise on this machine isn't very threatening due to the fact it's a workstation |
| System Countermeasures | 4 | Machine has most updated OS with all relevant service packs. |
| Network Countermeasures | 3 | Router and firewall weren't an issue since it was a lab environment |
| **Total** | **1** | |

**Detect #5 (produced in lab)**

| | | | |
|---|---|---|---|
| 00:13:19.532942 | attacker.13502 | > lab.1.513: | S 2112342080:2112342080 (0) win 32120 (DF) |
| 00:13:19.532945 | attacker.13503 | > lab.2.513: | S 2107900681:2107900681 (0) win 32120 (DF) |
| 00:13:19.532945 | attacker.13503 | > lab.3.513: | S 2109761710:2109761710 (0) win 32120 (DF) |
| 00:13:19.532946 | attacker.13504 | > lab.4.513: | S 2102386733:2102386733 (0) win 32120 (DF) |
| 00:13:19.532948 | attacker.13505 | > lab.4.513: | S 2102386733:2102386733 (0) win 32120 (DF) |
| 00:13:19.532949 | attacker.13506 | > lab.5.513: | S 2112342080:2112342080 (0) win 32120 (DF) |
| 00:13:19.532950 | attacker.13508 | > lab.2.513: | S 2107900681:2107900681 (0) win 32120 (DF) |

**Synopsis of Detect**

This appears to be an automated SYN scan on port 513. Port 513 is the rshell port on UNIX hosts. A compromise of one of these machines could be costly.

| **Issue** | **Score** | **Reason** |
|---|---|---|
| Criticality | 4 | None of the machines are critical machines |
| Lethality | 5 | Unix machines are very powerful, especially if root access is obtained |
| System Countermeasures | 4 | Machines have fairly new version of OS |
| Network Countermeasures | 4 | Router and Firewall will not permit connection using this port on any of these machines |
| **Total** | **1** | |

**Detect #6**

```
15:32:11.509142 mikhail.com > zhivago: icmp: echo request
15:32:11.509198 zhivago > mikhail.com: icmp: echo reply
15:32:11.509481 mikhail.com > zhivago: icmp: echo request
15:32:11.509514 zhivago > mikhail.com: icmp: echo reply
15:32:11.509788 mikhail.com > zhivago: icmp: echo request
15:32:11.509820 zhivago > mikhail.com: icmp: echo reply
15:32:11.510094 mikhail.com> zhivago: icmp: echo request
15:32:11.510128 zhivago > mikhail.com: icmp: echo reply
15:32:11.510401 mikhail.com > zhivago: icmp: echo request
15:32:11.510432 zhivago > mikhail.com: icmp: echo reply
15:32:11.510708 mikhail.com > zhivago: icmp: echo request
15:32:11.510739 zhivago > mikhail.com: icmp: echo reply
15:32:11.511014 mikhail.com > zhivago: icmp: echo request
15:32:11.511046 zhivago > mikhail.com: icmp: echo reply
15:32:11.511318 mikhail.com > zhivago: icmp: echo request
```

**Synopsis of Detect**

This is a ping flood, used possibly as a DOS attack. This came from a Linux host running the ping –f command.

| **Issue** | **Score** | **Reason** |
|---|---|---|
| Criticality | 3 | This is not a critical machine |
| Lethality | 2 | The worst that could happen is no use of the machine |
| System Countermeasures | 4 | Machines have fairly new version of OS and patches |
| Network Countermeasures | 4 | Router and Firewall will not permit connection using this port on any of these machines |
| **Total** | **-3** | |

**Detect #7**
*Apr 1 5:31:17.302: %SEC-6-IPACCESSLOGP: list 101 denied tcp attacker.9(1057) -> hometown.31337(4459), 1 packet
*Apr 1 5:31:21.493: %SEC-6-IPACCESSLOGP: list 101 denied tcp attacker.9(1048) -> hometown.65322(4452), 1 packet
*Apr 1 5:33:12.364: %SEC-6-IPACCESSLOGP: list 101 denied tcp attacker.9(1091) -> hometown.20631(135), 1 packet
*Apr 1 5:36:45.091: %SEC-6-IPACCESSLOGP: list 101 denied tcp attacker.9(1096) -> hometown.43922(4459), 5 packets
*Apr 1 5:37:25.445: %SEC-6-IPACCESSLOGP: list 101 denied tcp attacker.9(1048) -> hometown.58772(4459), 3 packets
*Apr 1 5:38:19.120: %SEC-6-IPACCESSLOGP: list 101 denied tcp attacker.9(1057) -> hometown.61535(4452), 5 packets

**Synopsis of Detect**
Initially, I suspected a DOS attack on hometown. But, after further investigation, I realized the attacker made multiple attempts to connect to the same machine via high (unused) port. This leads me to believe it was an attempt to connect to hometown using Back Orifice.

| **Issue** | **Score** | **Reason** |
|---|---|---|
| Criticality | 5 | Hometown is our DNS server |
| Lethality | 5 | If access was obtained, our DNS (and possibly more) would have been jeopardized |
| System Countermeasures | 4 | Machines have fairly new version of OS |
| Network Countermeasures | 4 | Router and Firewall will not permit connection using this port on any of these machines |
| **Total** | **2** | |

**Detect #8**
*Apr 3 17:54:29.052: %SEC-6-IPACCESSLOGP: list 101 denied tcp attacker.1323(1091) -> dotcom.31337(135), 3 packets
*Apr 3 17:54:29.414: %SEC-6-IPACCESSLOGP: list 101 denied tcp attacker.1323(1048) -> dotcom.65322(4459), 1 packet
*Apr 3 17:55:41.230: %SEC-6-IPACCESSLOGP: list 101 denied tcp attacker.1323(1057) -> dotcom.20631(135), 1 packet
*Apr 3 17:56:42.309: %SEC-6-IPACCESSLOGP: list 101 denied tcp attacker.1323(1096) -> dotcom.43922(4452), 5 packets
*Apr 3 17:57:09.112: %SEC-6-IPACCESSLOGP: list 101 denied tcp attacker.1323(1048) -> dotcom.58772(4459), 1 packet
*Apr 3 17:57:57.120: %SEC-6-IPACCESSLOGP: list 101 denied tcp attacker.1323(1096) -> dotcom.61535(4452), 3 packets
*Apr 4 17:58:43.529: %SEC-6-IPACCESSLOGP: list 101 denied tcp attacker.1323(1057) -> dotcom.31337(4459), 5 packets
*Apr 4 17:59:16.356: %SEC-6-IPACCESSLOGP: list 101 denied tcp attacker.1323(1048) -> dotcom.65322(135), 1 packet
*Apr 4 17:59:55.098: %SEC-6-IPACCESSLOGP: list 101 denied tcp attacker.1323(1091) -> dotcom.20631(4452), 1 packet

**Synopsis of Detect**
This appears to be a port scan on a specific host that was detected by our router log. This may have been crafted, because of the same source port, but the gap in time makes me think it was manual as opposed to automated.

| **Issue** | **Score** | **Reason** |
|---|---|---|
| Criticality | 5 | Dotcom is our web server |
| Lethality | 5 | If this machine is compromised, all of our web apps will be dead |
| System Countermeasures | 4 | Machine has new version of OS and adequate patches |
| Network Countermeasures | 4 | Router and Firewall will not permit connection using any of these ports on this machine |
| **Total** | **2** | |

**Detect #9 (produced in lab)**

```
19:52:21.271032 malicious.4321 > helpme.9876: udp 10
22:52:21.271032 malicious.4321 > helpme.9876: udp 10
22:52:21.271032 malicious.4321 > helpme.9876: udp 10
22:52:21.271032 malicious.4321 > helpme.9876: udp 10
22:52:21.271032 malicious.4321 > helpme.9876: udp 10
```

**Synopsis of Detect**

This appears to be a upd flood on a machine. I don't think this is a very malicious detect, maybe even an accidental incident. There is no history (though our history log is short) and we haven't seen any traffic from malicious since this incident.

| **Issue** | **Score** | **Reason** |
|---|---|---|
| Criticality | 3 | This particular machine isn't a critical machine |
| Lethality | 3 | It's a windows host, so even if it were compromised, it wouldn't be very crucial |
| System Countermeasures | 4 | Machine has new version of OS (and service packs) along with virus protection software |
| Network Countermeasures | 4 | Router and Firewall will not permit connection using this port on any of these machines |
| **Total** | **-2** | |

**Detect #10 (produced in a lab)**

```
14:51:19.520937 zhivago.10 > zhivago.10: udp 28 (frag 1109:36@0+)
14:51:19.520942 zhivago > zhivago: (frag 1109:4@32)
14:51:19.520991 zhivago.11 > zhivago.11: udp 28 (frag 1109:36@0+)
14:51:19.520997 zhivago > zhivago: (frag 1109:4@32)
14:51:19.521003 zhivago.12 > zhivago.12: udp 28 (frag 1109:36@0+)
14:51:19.521009 zhivago > zhivago: (frag 1109:4@32)
14:51:19.521018 zhivago.daytime > zhivago.daytime: udp 28 (frag 1109:36@0+)
14:51:19.521024 zhivago > zhivago: (frag 1109:4@32)
14:51:19.521075 zhivago.14 > zhivago.14: udp 28 (frag 1109:36@0+)
14:51:19.521080 zhivago > zhivago: (frag 1109:4@32)
14:51:19.521086 zhivago.15 > zhivago.15: udp 28 (frag 1109:36@0+)
14:51:19.521128 zhivago > zhivago: (frag 1109:4@32)
```

**Synopsis of Detect**

This is a Teardrop attack. I caught this while running a tdcpdump and noticed that the attack originated from the host it was destined for. This is obviously a script because of the proximity of the fragments with respect to time. This attack was on a UNIX host, and it withstood the attack.

| **Issue** | **Score** | **Reason** |
|---|---|---|
| Criticality | 4 | This unix machine is just a normal host in our lab |
| Lethality | 3 | The only thing that would happen is a potential DOS on this machine |
| System Countermeasures | 4 | Machine has new version of OS and patches |
| Network Countermeasures | 4 | Router and Firewall will block excessive traffic to this machine (outside the lab) |
| **Total** | **-1** | |