EN GARDE SYSTEMS
INCORPORATED

Sample Test Report

Internet Penetration Test

Customer XXX

CONFIDENTIAL

PREPARED BY:

EN GARDE SYSTEMS, INC.
4848 TRAMWAY RIDGE DR. NE SUITE 122
ALBUQUERQUE, NM 87111
(505) 346-1760

June 29, 2000

Table of Contents

Overview

Executive Summary

Each of our reports is highly tailored to the individual customer but will generally follow the following outline.

Testing Process and Procedures

En Garde Systems (EGS), Incorporated performs penetration testing as the primary means to evaluate overall system and network security. All testing is performed in accordance with the testing protocol in Appendix A. The tools, methods, and techniques employed by EGS to perform these tests are generally well known throughout both the computer security and “hacker” communities. Hence, vulnerabilities or configuration liabilities discovered as a result of these tests can be viewed as those that any intruder may find while testing the network and connected systems. Tests were conducted over the Internet to determine if external network security controls (firewalls, servers, routers, etc.) are effective in preventing unwanted external intrusion, and unauthorized access XXX resources on the internal network. All Internet tests were accomplished from the perspective of an outsider[1] trying to gain unauthorized access. Tests were also conducted from a EGS test station on the internal network to determine the risk from insiders gaining more access than they are allowed.

[1] An outsider is someone who has no authorized access to the data or systems to which they wish to login. These are the persons commonly referred to as "hackers" as their goal is to gain inside access by utilizing some form of security override.

Test Results

Public Exposure

InterNIC inquiries and DNS Zone Transfers are frequently employed by hackers, business competitors, or industrial spies to collect useful information about a specific network or domain. Each of these methods and techniques was used by EGS to collect information concerning domains registered to XXX.

InterNIC Queries

The registration information for the networks being tested is listed below.

```
En Garde Systems (NET-ENGARDE)
   525 Clara Ave. #202
   St. Louis, MO  63112

   Netname: ENGARDE
   Netblock: 199.165.219.0 - 199.165.219.255

   Coordinator:
      Neuman, Michael  (MN33-ARIN)  mcn@ENGARDE.COM
      314-578-1894

   Domain System inverse mapping provided by:

   NS.ENGARDE.COM               199.165.219.1
   ARIEL.SDSC.EDU               132.249.22.240

   Record last updated on 19-Oct-1995.
   Database last updated on 26-Feb-2001 06:34:32 EDT.
```

DNS Zone Transfers

Domain name servers can sometimes be used to obtain information about hosts on a given network through a technique called zone transfer. Zone transfers are commonly used to synchronize the primary and secondary name servers. When properly configured, the servers should normally allow only the secondary name server to obtain a zone transfer; under no circumstances should the name servers be configured to zone transfer information to the Internet.

The following information was obtained in a zonetransfer from ns.engarde.com:

```
; <<>> DiG 8.1 <<>> @ns.engarde.com axfr engarde.com
; (1 server found)
$ORIGIN engarde.com.
@                 1D IN SOA   ns postmaster.ns (
                              9005322           ; serial
                              12H         ; refresh
                              5M          ; retry
                              5w6d16h           ; expiry
                              1D )        ; minimum

                  1D IN NS    ns
                  2H IN MX    0 mh1dmz1
                  2H IN MX    0 mh2dmz1
ctweb1                  1D IN A           199.165.219.53
ctweb2                  1D IN A           199.165.219.57
ads               5M IN NS    ds1-2b
                  5M IN NS    ds1-da
farm        1D IN A           199.165.219.58
tv1               1D IN CNAME y881
iwos              1D IN CNAME farm
wp1               1D IN A           199.165.219.59
www4              1D IN A           199.165.219.48
```

Recommendation: Disallow zone transfers to random Internet hosts. This activity should only be allowed to secondary name servers.

Configuration Analysis

We were given a partial list of the IP-addresses assigned to EGS. We scanned each machines found on the 199.165.219.0 network using: Internet Control Message Protocol (ICMP), Transmission Control Protocol (TCP), and the User Datagram Protocol (UDP). The results of the service scans are shown in Table II-1.

Table II-1. Scan Results

Machine	TCP Port	Service
199.165.219.37	25	"220 mh21.engarde.com (IntraStore TurboSendmail) ESMTP Service ready"
199.165.219.36	25 80	"220 mh11.engarde.com (IntraStore TurboSendmail) ESMTP Service ready" HTTP- "Microsoft-IIS/5.0"
199.165.219.35	21	"220 bfmdr FTP server (DG/UX Release

	2766	R4.20MU06) ready."
199.165.219.25	21 1030 2010 2766	"220 bfm FTP server (DG/UX Release R4.20MU06) ready."
199.165.219.102	21 2010 2766	"220 ftp1 FTP server (DG/UX Release R4.20MU04) ready."

The ftp servers are running too many services. Each service that is open on a machine allows a possible point of entry for an external attacker.

Recommendation: The unneeded open ports on the ftp servers should be turned off or filtered out, including: 1030, 2010, and 2766.

We were able to traceroute and ping any internal machine on the EGS network. This access allows a number of denial of service attacks to reach internal machines.

Recommendation: If possible this access should be turned off.

The firewalls are also configured so that TCP port 0 is not blocked to the DMZ. There are a number of denial of service attacks which send packets to port 0.

Recommendation: Filter out packets sent to port 0.

Internet Penetration Test

Network Level Tests

An entire class of attacks are aimed at circumventing the distinction between the inside and outside of a firewall or gateway routers to gain access to internal network resources. Common attacks and specific test results are summarized below (results applicable to the EGS network are in bold italics):

(a) **Source Routing**. By spoofing the source address and adding a “Source Route” entry into the IP packet, it is possible to contact a target, make it believe the packet came from the spoofed source computer, and yet send responses back to the source routing computer. Common attacks use the source routing computer as the attacking system so that replies to the attacks are sent directly back to the attacker. ***Source routed packets appear to be blocked by the ISP.***

(b) **Internet Control Message Protocol (ICMP) Bombing.** By sending an ICMP host unreachable (or net unreachable) packet to the router, an attacker attempts to deny service to specific internal connections. Most systems will blindly accept a "host unreachable" message from any system and instantly terminate every connection to the host referenced in the ICMP message. If the attack is successful, the router will accept host unreachable messages from the outside and terminate internal connections, thus causing a denial of service disruption***. ICMP unreachable messages were not attempted for fear of disrupting service. ICMP traffic should be filtered out at the router or firewall if not needed.***

(c) **Sequence Number Prediction**. The Transmission Control Protocol (TCP) sequence number is predictable, and hosts may be trusted to varying degrees. As a result, it is possible for an intruder, or malicious insider to spoof connections from "trustworthy" addresses. ***Not specifically tested during the external test; sequence number prediction requires knowledge of inter-machine trust relationships, on the open Internet ports.***

(d) **Routing Information Protocol** (RIP). By sending RIP messages, it is possible to change a host's routing tables. ***Not specifically tested since we did not want to disrupt service on the production machines. This attack has to be sent from the local network.***

(e) **Address Resolution Protocol (ARP) Cache Problems**. By sending gratuitous ARP replies, it's possible to change the physical address to IP address mapping of a system. ***Not specifically tested since we did not want to disrupt service on the production machines. This attack has to be sent from the local network.***

(f) **"Sniffing" or Network Monitoring, and TCP Hijacking.** If encryption is not supported on a network, any connection can be monitored by an intruder who successfully installs a "sniffer"[2] on a router, server, or even the firewall (note: insiders do not have to compromise a network device, since they can simply install the sniffer on their own computer). Once the sniffer is installed, the intruder has access to all the information passing over the connection including passwords. Monitored connections can be hijacked if the intruder is positioned on the network between the client and the server of the connection. The connection can be initiated from inside or outside the firewall. The connection can also be authenticated by a smartcard or other means of one time authentication, but

[2] A "sniffer" is a program commonly used by hackers to monitor connections and steal passwords.

since TCP hijacking can take over an unencrypted connection after it is authenticated, the authentication will make no difference. ***All authentication data sent from the Internet to ftp, HTTP and other unencrypted connections can be sniffed, captured, and used to hijack EGS systems.***

(g) **Flooding TCP Service Queues**. TCP has a finite set of states with well-defined transitions between each state. The first packet ("SYN") forces the receiver into "SYN RECVD" state, whereby it responds with a "SYN"/"ACK" packet. The receiver then waits for a long period of time (30-60 seconds) for the original sender to respond to the "SYN"/"ACK" packet. If the original sender never responds, the receiver closes the connection and returns to the "LISTEN" state. Many systems only allow a finite number of "SYN RECVD" connections. If any more packets are received after that number of connections are in "SYN RECVD", they are simply dropped. As a result, an intruder can turn off specific protocols. ***All hosts on the DMZ that offer TCP services to the Internet are possibly vulnerable to TCP flooding (denial of service) attacks; however, such tests were not attempted for fear of disrupting operations.***

Service Level Tests

Another class of attacks are directed at defeating individual services offered by a system on the network. Each of the services offered by a machine was tested to determine if it could be used to pass data to the inside, accept dangerous commands, or have its access controls otherwise circumvented to compromise the machine on which it was running. Results of these tests are summarized below.

General recommendations for all services running on a network:

1. If the service is not being used it should be turned off.
2. The latest version of the service should be installed to avoid known vulnerabilities.

The following services were being run on the network:

(a) ***Sendmail***. (TCP port 25) Sendmail is probably the most historically unsecured service, and there are many different ways to exploit older versions of this program, including buffer overflows, race conditions, and Domain Name Service (DNS) spoofing. In addition, sendmail can sometimes be used to identify machines on the internal network, if the server relays mail requests anywhere on the internal network and returns internal IP addresses to the sender.

The sendmail servers are well configured to filter out malicious e-mail, and updated to eliminate known security vulnerabilities.

(b) ***FTP.*** (TCP port 21) The FTP protocol is complex in that it must accept incoming connections, as well as initiate outgoing connections to carry transmitted data. As a result of the complexity, the FTP protocol is difficult to secure. We found several vulnerabilities in the FTP configuration. Solutions and fixes to each of these problems were being put in place when we left.

(1) Anonymous Access allowed to FTP servers. Each of the FTP servers on the DMZ allowed Anonymous access. This allowed any user on the Internet to gain access to the server and using the combination of vulnerabilities listed below, they could then gain access to the Internal network.

Recommendation: Anonymous access is not needed and should be turned off.

(2) *Anonymous FTP Directory Ownership**.*** Many configuration manuals suggest making the anonymous FTP directory owned by the user ‘ftp”. However, if the directory is owned by FTP, the anonymous FTP user could insert a file under any name into the FTP user’s home directory. New binaries could be uploaded to replace standard ones (such as uploading a shell to replace ~ftp/bin/ls).

Recommendation: While working with the EGS staff, the solution of making the FTP directories owned by root but with the sticky bit set was found to both eliminate the vulnerability and also allow customers to upload files into the root directory.

Since we had gained access to the DMZ from the Internet but did not want to disrupt service, we were given an account on ftp1 and allowed to test the security of the Internal network assuming that the DMZ had been compromised.

We were able to find a number of SetUID scripts written by EGS staff. These make ideal targets, and we were able to exploit one of these scripts to gain root access to ftp1. The specific script exploited was /example/test1 which when run by us gave an error stated it could not run /example/test2.sh. We then changed the Internal File Separator variable to “/” in our local environment and created a program in our local directory called example. This change made the test1 program run our local example program instead of test2. This allowed us to run any command as root. In this case we changed the permissions on a program that changed the users UID to root so that it could be run by any user. This gave us a root shell on the machine.

Recommendation: Carefully review each SetUID script on the critical systems for vulnerabilities that could allow a normal user to gain root access.

(c) ***HTTP and HTTPS.*** (TCP ports 80 and 443 respectively)

Many times server side web pages are created with vulnerabilities inherent in them. We also check for architecture problems in terms including transmitting information in the clear when it should be encrypted and problems inherent in the web server.

On www.engarde.com, 199.165.219.36, we found a cgi script, /cgi-bin/feedback.cgi which allowed any file in the /web/feedback directory to be viewed by Internet users. It had apparently been patched earlier to keep people from opening any file in any directory on the system; but if possible the files to be opened should be limited to a specific set.
Recommendation: Limit the files, which can be opened to expected entries.

This specific script also allowed any command to be run on the system by simply putting the command in the e-mail address line. By specifying an e-mail address of

‘ls –l > /web/feedback/blah.txt’@xyz.com

we were able to see the listing of the current directory. The same format can run any system command and write the results to a file or send the attacker e-mail.

We also created a simple perl script, which opened up a port on the web server and created a shell on the web server when we connected. It was possible to find a port not blocked by the firewall, by TCP scanning the system, so we used port 1080 for the backdoor. We made sure it was un-installed before exiting the system.

Recommendation: Make sure that any user input that which is sent to a system call, open call or any other shell environment has had all malicious input removed. The specific characters that the user is allowed to enter should be specified in the script. It is much better to specifically allow characters rather than specifically deny bad characters. Additionally if alarms can be built into the system to notify security administrators when bad characters are being tested security incidents can be identified and fixed before any harm has come to the web servers. While we were on-site this particular script was patched on this machine.

Shell meta-characters ([;<>*|`&$!?#(){}:'"\^]) are very dangerous, since they can allow the hacker unlimited access to the machine. They have the capability to spawn a shell or perform operations on sensitive files. In order to avoid these vulnerabilities, the safe characters must be specified and the script or code must reject all other characters. Filtering the meta-characters is better than simply escaping them, as a hacker can easily avoid an escape.

While we were on the system we discovered that older versions of this script were still on the system. These older versions allow hackers to exploit security problems that have been fixed in the most recent versions.
Recommendation: Make sure all default, old, and unused scripts are removed from the servers.

Recommendations

General Comments

The following is a summary of the recommendations made during the test.

Recommendation: Disallow zone transfers to random Internet hosts. This activity should only be allowed to secondary name servers.

Recommendation: The unneeded open ports on the ftp servers should be turned off or filtered out, including: 1030, 2010, and 2766.

Recommendation: If possible icmp packets should be filtered out.

Recommendation: Filter out packets sent to port 0.

Recommendation: Anonymous access to the FTP servers is not needed and should be turned off.

Recommendation: While working with the EGS staff, the solution of making the FTP directories owned by root but with the sticky bit set was found to both eliminate the vulnerability and also allow customers to upload files into the root directory. This should be implemented across the network

Recommendation: Carefully review each SetUID script on the critical systems for vulnerabilities that could allow a normal user to gain root access.

Recommendation: Limit the files, which can be opened by web scripts to expected entries.

Recommendation: Make sure that any user input that which is sent to a system call, open call or any other shell environment has had all malicious input removed. The specific characters that the user is allowed to enter should be specified in the script. It is much better to specifically allow characters rather than specifically deny bad characters. Additionally if alarms can be built into the system to notify security administrators when bad characters are being tested security incidents can be identified and fixed before any harm has come to the web servers. While we were on-site this particular script was patched on this machine.

Recommendation: Make sure all default, old, and unused scripts are removed from the servers.

Testing Protocol

Tools

Some of the tools used to conduct probes and tests are proprietary. These tools are used to test:

- Partial packets.
- TCP port flooding.
- Source routing.
- Sequence number prediction.
- Certain forms of routing attacks (RIP, EGP, etc.)

Many other tools are freely available on the Internet. Some of these are described below; most can be found @ ftp://coast.cs.purdue.edu/pub/tools/unix:

- ***Tklned/Scotty*** -- A tool for mapping and monitoring network segments. This tool is used to discover hosts on the network and automatically generate a map of the network. It can also be used to perform a number of SNMP queries to determine subnetting and ARP caches from the router.
- ***Strobe*** -- A port scanner; By modifying this tool slightly, it will not cause 2+ minute timeouts on each port scanned.
- ***Newscan*** -- A half-open port scanner. By not completely opening ports, logging is completely ineffective. As a result, ports can be scanned without tripping alarms.
- ***TCPdump*** -- A low level network monitoring tool used to monitor and record the responses to certain types of firewall probes.

Test Methodology

In general, the tools described above, and other techniques, are used to develop basic information about a network, and any protection measures that are in place, to: (i) characterize the system security contour; (ii) determine network vulnerabilities/weaknesses, and configuration or system liabilities; and (iii) carry out network and service layer attacks. For example:

DNS. Basic DNS inquiries are accomplished; attempts are made to zone transfer information about the network.

Tracerouting is used to gather additional information about internal hosts and IP addresses to determine possible scenarios for routing attacks.

Network Scanning. The network is scanned to determine specifically what can be seen from outside; TCP/IP queries are made in an attempt to gather internal network information; a standard TCPscan of hosts will determine what services are being offered by network devices.

Access Granted. Attempts are made to subvert network or host access controls and/or authentication methods. Attacks can be as simple as guessing bad passwords, or as complex as subverting available services. Examples of some (not inclusive) attacks at both the network and service levels are described below:

Network Layer Attacks

A whole class of attacks are directed at circumventing any distinction between the ***inside*** and ***outside*** of the network as might be defined by a firewall, screening router, or other device. Before attempting to penetrate individual services, attempts are made to confuse the networking implementation to give more access than authorized by policy:

- ***Sequence Number Prediction***. By spoofing the source address and attempting to predict the TCP sequence number (which a firewall will send in reply to our spoofed packet), a full TCP connection can be started from a putative address. If successful, the firewall will pass packets from the false address to the inside of the firewall. Another possible attack is if the firewall believes the incoming packet's address is from the inside even though it's being received from an external daemon.
- ***ICMP Bombing***. By sending an ICMP host unreachable (or net unreachable) packets to a specific host, service can be denied to specific internal connections. Most systems will blindly accept a "host unreachable" message from any system and instantly terminate every connection to the host reference in the ICMP message.
- ***Source Routing***. By spoofing the source address and adding a "source route" entry into the IP packet, it is usually possible to contact a target, make it believe the packet came from the spoofed source computer, but still send its responses back to the source routing computer. Common attacks have the source routing computer be the attacking system so that replies to the attacks are sent directly back to the intruder.
- **Routing Information Protocol (RIP).** By sending RIP messages, it is possible to change a host's routine tables.
- ***ARP Cache Problems***. By sending gratuitous ARP replies, its possible to change the physical address to IP address mapping of a system.
- **"Sniffing" or Network Monitoring**. By monitoring all traffic on a network, its possible to record users' passwords as they type them to login to a system.

- ***TCP Hijacking.*** By monitoring and inserting packets into a network, it is possible to steal a connection from users who have already authenticated themselves.
- ***IP Fragmentation Overwriting***. Because IP can be fragmented to travel over networks with smaller packet sizes, its often the case where a packet is split up into several pieces and reassembled on the destination host. Many firewalls make decisions about whether data should be allowed to pass based upon whether or not the TCP "SYN" bit is set with no TCP "ACK" bit. If a "SYN"-only packet is received, it indicates a new connection is being established from the outside of the firewall. On the other hand, if a "SYN"/"ACK" packet is received, it means a host internal to the firewall is attempting to start a connection. Filter rules based upon these bits is often important if you wish to allow outgoing connections, but not incoming ones. According to the Internet RFCs, the first fragmented packet of a TCP/IP connection must be large enough to include the TCP flags. As a result, firewalls may filter on the first packet alone. Unfortunately, it is possible that if the IP packet is fragmented, the first fragment contains the "SYN"/"ACK" flags, and the second contains data which overwrites the original's TCP flags with a simple "SYN". Because the firewall is filtering out the first case, the second packet is ignored. However, on the destination host, when the packet is reassembled, it appears as a "SYN" only packet, thus starting a new connection.
- ***Flooding TCP Service Queues***. TCP is a stateful protocol. The first packet (a "SYN") forces the receiver into "SYN RECVD" state, whereby it responds with a "SYN"/"ACK" packet. The receiver then waits for a long period of time (30-60 seconds) for the original sender to respond to the "SYN"/"ACK" packet. If the original sender never responds, the receiver closes the connection and returns to the "LISTEN" state. The problem here is that many systems only allow a finite number of "SYN RECVD" connections. If any packets are received after that number is reached, they are simply dropped off. As a result, specific protocols can be turned off by anyone.

Service Layer Attacks

Another class of attacks are directed at defeating individual services offered by a system on the network. If a service can be convinced to pass data to the inside, accept dangerous commands, or have its access controls otherwise circumvented, internal machines are vulnerable to attack:

- ***Sendmail.*** Sendmail is probably the most historically unsecured service, with many ways to exploit older versions of this program.
- ***Finger Buffers***. Some finger daemon implementations have a bug in which sending too much data will overflow the input buffer resulting in that data being placed directly on the stack. It then is relatively trivial to force the finger

daemon to directly execute the instructions which were forced on the stack. In this way, a machine running a buggy finger daemon can be forced to execute arbitrary instructions. This is one of the attacks used by the Morris/Internet Worm about five years ago.

- ***HTTPD Buffers.*** Programmers tend not to learn by their mistakes. The most common version of the HTTP daemon had the exact same bug as the finger daemon once had. Input lengths were not appropriately truncated. Consequently, the HTTP daemon would execute arbitrary instructions.
- ***NFS/Mountd.*** The NFS and Mount daemons are fraught with security holes. We attempt to exploit many of then, including file handle guessing, export field length overflows, and uid mapping bugs.
- **Portmapper Indirect Calls.** The default portmapper shipped with many Unix systems will forward requests to service daemons directly to increase efficiency. Unfortunately, many of these service daemons trust the machine they are running on. As a result, when the portmapper forwards the service request, it appears to the daemon that the request is coming from localhost and grants access.
- ***TFTP.*** Although useful, this protocol has many possible security vulnerabilities.
- ***Anonymous FTP Directory Ownership.*** Many configuration manuals suggest making the anonymous FTP directory owned by the user 'ftp". However, if the directory is owned by FTP, the anonymous FTP user could insert a file under any name into the FTP user's home directory. This file could be ".rhosts", "login", or even a .plan linked to the/etc/ passwd file, so when the ftp user is fingered, the password file is displayed. In addition to adding administrative files, new binaries could be uploaded to replace standard ones (such as uploading a shell to replace ~ftp/bin/ls).
- ***Real Passwords in the Anonymous FTP passwd File.*** When configuring an anonymous FTP directory, many configuration manuals have the user simply copy the real password file into the anonymous FTP directory structure. This allows any anonymous user to retrieve the real passwd file and run a cracking program on it.
- ***NNTP.*** Many of the newer NNTP servers simply pass control messages to the mail program or to the shell. A malicious user can insert mail (or shell) escape sequences into forged control messages, causing the server to execute arbitrary instructions.
- ***FTP Service.*** The FTP protocol is complex in that it must accept incoming connections, as well as initiate outgoing connections to carry transmitted data. As a result of the complexity, the FTP protocol is difficult to secure.
- ***Xwindows Proxy.*** If internal hosts are allowed to start X Window clients on remote machines, the access control granted to the remote machine must be

carefully monitored. Several commonly available programs allow the remote host to monitor (and even insert data) an internal host that has started an X client.

- ***Fingerback.*** Many Host systems implement a finger back trap to catch suspicious users in the act. When a suspicious action is detected, the host will automatically finger the remote machine. Several problems result from this -- the data which is saved from the remote machine may not have a length limit, resulting in the remote user sending infinite data streams back to the host; or the remote host may finger back to the firewall, which could cause an infinite loop as the firewall fingers back.
- ***External DNS Zone Xfers.*** Retrieve internal DNS from the outside.
- ***FTP "site exec sh-c id".*** A recent version of the FTP daemon allows specific commands to be executed. However, many daemons are not very careful about which commands the user specifies.

The Process

Testing is accomplished using a process and methods proven effective by the National Security Agency (National Computer Security Center). This closed loop process is shown in the table below:

STEP NUMBER	DESCRIPTION
1	**System Security Contour.** Perform an inventory of configuration items at the network level. Conduct a functional allocation trace to allocate system security processes with network-level components, interfaces, protocols, and security requirements.
2	**System Vulnerability Examination.** Research potential network vulnerabilities. System security contour data is used here to "narrow" the potential vulnerability search and focus the generation of an attack plan. Develop a baseline network architecture and identify potential network weaknesses and vulnerabilities.
3	**Paper Penetration.** Map the network, its components, and the protocols in use to the potential vulnerabilities identified in step 2. Conduct a target analysis to allocate vulnerabilities to a component or protocol and develop attack scenarios.
4	**Tools Analysis.** Research and select tools on the basis of their ability to provide the functionality needed to carry out attack scenarios.
5	**Penetration Attack.** Conduct penetration attacks using tools

	selected in step 4, in accordance with attack scenarios developed in step 3. Prepare attack results report to address actual network vulnerabilities yielded by the attack.
6	**Vulnerability Analysis.** Assess risks and formulate countermeasures or other actions that can be taken to mitigate risks for each network vulnerability or weakness discovered as a result of penetration tests.
7	**Feedback Process.** Update the system security contour (baseline), vulnerability database, and the tools catalog with the results of the attack scenarios.

En Garde Systems would like to thank you for your business. Aside from benefiting from the highest level of service possible, you also received unique advantages that only En Garde Systems delivers.

Experience Counts

While EGS maintains a low profile in the computer security market, we are the most trusted advisors to some of the largest companies in the world. We often work directly with upper-management, even providing ideas and input on a daily basis in some cases. Frequently, we are asked to review the work of other contractors and make final recommendations to management. We have provided security solutions to a diverse list of clients ranging from Fortune 500 companies to small businesses. Our customers come from many fields, including government, military, law enforcement, and intelligence in the public sector. Our private sector clients have ranged from the transportation and energy fields to the communications, insurance, and financial sectors.

We enjoy an excellent reputation within the computer security community and garner glowing references from our customers. In fact, over 30% of our business is derived from repeat customers.

The “Human Touch”

Our experience has revealed undetected security issues in professionally run sites. EGS has performed "follow-up" tests after some of our larger competitors, including “Big Six” accounting/auditing firms. In most instances, we found problems their automated tests missed.

EGS pioneered the hands-on approach to penetration testing. Instead of relying on commercial test packages like everyone else, we put our brains to work. Hackers don’t use expensive test tools to break into your network—and neither do we. We analyze each network for new and unique vulnerabilities and develop tools specific to you. We do our own research and write our own tools.

Our hands-on approach is tailored to fit both the needs of your company and of your networks. After all, no two companies are the same, and no two networks are, either. Tools written to examine generic networks ignore the constantly changing security environment of your company. An automated “cookie-cutter” approach makes it impossible to effectively test every potential problem. By not simply relying on automated tools to do the work for us, we offer evaluations that are more thorough than those you would receive from our competition.

A Better Deal

Our recommendations rarely require our customers to buy new equipment. In our considerable experience, we have discovered that most security problems can be fixed without spending any money at all. Unlike our competitors, we won’t suggest purchasing expensive new hardware or software you don’t need. En Garde Systems’ unique, hands-on approach allows us to make recommendations on a per customer basis. If it is necessary to buy additional products, we will recommend several different approaches specific to your network that vary in price.

Our technical expertise, outstanding reputation, and personalized attention ensure you a level of service surpassed by no other security firm in the market. As an EGS customer, you can be confident in your sound decision to improve your network security!