White Paper

Protecting Your Critical Assets

Lessons Learned from "Operation Aurora"

By McAfee Labs and McAfee Foundstone Professional Services

Table of Contents

Executive Summary

As Operation Aurora highlighted, advanced persistent threats (APT) are an increasingly common form of complex and directed attacks that use insidious techniques for gaining access to privileged systems and maintaining that access until all of the attackers' goals and objectives have been met. Operation Aurora employed an APT technique that proved extremely successful in targeting, exploiting, accessing, and exfiltrating highly valuable intellectual property from its victims. This paper details Operation Aurora and provides some insight into what was learned and how to prevent such attacks from being successful in the future.

How Aurora Worked

Operation Aurora included numerous steps that all occurred invisibly in an instant from the user's perspective. As you can see in the illustration below, without any apparent signs of malicious intent or actions, Operation Aurora completed its attack in six simple steps:

1. A targeted user received a link in email or instant message from a "trusted" source.
2. The user clicked on the link which caused them to visit a website hosted in Taiwan that also contained a malicious JavaScript payload.
3. The user's browser downloaded and executed the malicious JavaScript, which included a zero-day Internet Explorer exploit.
4. The exploit downloaded a binary disguised as an image from Taiwan servers and executed the malicious payload.
5. The payload set up a backdoor and connected to command and control servers in Taiwan.
6. As a result, attackers had complete access to internal systems. They targeted sources of intellectual property, including software configuration management (SCM) systems accessible by the compromised system. The compromised system could also be leveraged to further penetrate the network.

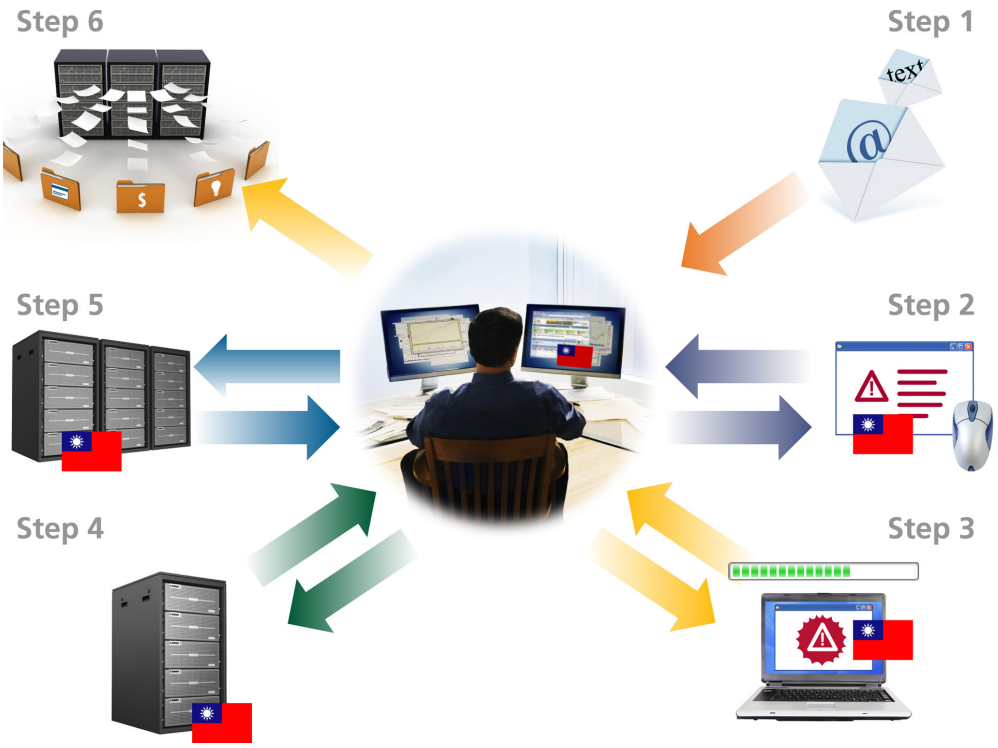


Figure 1. The complete steps of Operation Aurora's attack.

What We Learned

Directed, targeted attacks such as Operation Aurora are nothing new. Numerous attacks over the years have proven that attackers are calculated, patient, and stop at nothing to achieve their objectives, particularly when it comes to stealing sensitive information. But while the government, military, and defense industry base (DIB) have long been addressing these threats, the commercial entities targeted by Operation Aurora have been largely immune. But the picture has changed.

Intellectual Property

Numerous sources of intellectual property (IP) exist inside today's global companies, including trade secrets, proprietary formulas, copyrights, trademarks, and source code, to name a few. To say these IP sources represent the heart and core value of companies worldwide is an understatement. When these IP sources get compromised, capitalism and commerce are compromised on a global scale.

These intellectual property data stores come in many formats, including SCM systems like Perforce and IBM Rational Software and document/content management systems like Microsoft Sharepoint and EMC Documentum. Our experiences in managing the cleanup and assisting the affected companies have brought to light a new target of hacker attacks: IP repositories. To help protect our customers and the world as a whole, we directed our best and brightest to look at these systems and discover any vulnerabilities and weaknesses that could be exploited. During the coming months, we will expose this increasingly popular attack target and provide real countermeasures for this new, evolving threat.

Software Configuration Management (SCM)

Within large enterprises, source code is typically housed in source code trees within source code control systems, often called software configuration management (SCM) systems. Vendors include Perforce, Concurrent Versions System (CVS), Microsoft Visual Source Safe (VSS), IBM Rational, and others. In our experience, a number of these systems are not secured by default and typically rely on the customer to prevent an attacker from bypassing default controls. What are some of the most common threats to such a system?

- *Stealing a company's intellectual property by downloading the entire tree and then exfiltrating it out of the company's network*—This is a well-understood risk since it involves theft of intellectual property, which, as explained above, is often a technology vendor's core strength
- *Ability for both legitimate users as well as potentially unauthorized users to make changes to the tree*—This risk is especially significant since it allows an attacker to surreptitiously and discretely make changes to the source code that then go undetected and make their way into production releases
- *Utilizing the captured source code to find additional vulnerabilities in the affected products*—Attackers find it easier to launch new attacks against products when they have the blueprints used to build the software. This is a special case of intellectual property theft that puts the customers of the affected products at additional risk.

Insufficient out-of-the-box security mechanisms—plus the fact that organizations do not specifically lock down their SCM systems—give attackers unfettered access to these systems. What's more, obtaining this access is often trivial, given the lack of security controls. Because most SCM systems are not set to write and maintain sufficient logs to aid in investigation and/or recovery after an attack, the scenario can be particularly threatening to a company's IP store.

Why are these data points significant within the context of Operation Aurora? During the course of this investigation, we discovered numerous design and implementation flaws in a number of source code management systems that make cyberattacks highly viable.

Additionally, due to the open nature of most SCM systems today, much of the source code it is built to protect can be copied and managed on the endpoint developer system. It is quite common to have developers copy source code files to their local systems, edit them locally, and then check them back into the source code tree. As such, many code files can typically be found on endpoints such as the systems used by developer or quality assurance (QA) team members. As a result, attackers often don't

even need to target and hack the backend SCM systems; they can simply target the individual developer systems to harvest large amounts of source code rather quickly.

Intellectual Property Repositories Exposed

To empower our customers and commercial entities worldwide so that they can protect their sources of intellectual property, the following analysis (which is by no means all inclusive) can be considered the first wave of many pieces to emanate from McAfee® Labs™ about the security of the most popular systems housing intellectual property. In this paper, we look at Perforce on Microsoft Windows systems.

Perforce is a company in Alameda, California that has long been a staple of source code control systems and has thousands of customers (http://www.perforce.com/perforce/customers/byname.html). Its products are used by the largest Fortune 1,000 companies. We performed a security review of Perforce and uncovered some noteworthy findings.

Finding P-1—Perforce server service (p4s.exe) installs with "system"-level privileges

Common best practice for security is to provide the fewest privileges necessary to perform a given function. As such, it is generally recommended that software be installed as a limited user. Instead, Perforce runs its software as "system" under Windows. This practice gives malware the ability to inject itself into "system" level processes, giving access to all administrative functions on the system.

The Perforce documentation for UNIX directs the reader not to run the server service as root but never mentions making a similar change to the Windows service. As a result, the default installation runs as a local system which is just like running as root on Windows.

Finding P-2—Unauthenticated user creation

By default, unauthenticated anonymous users are allowed to create users in the Perforce depot. Additionally, no user password is required to create a user.

Finding P-3—Passwords are unencrypted

When installed, by default, Perforce transmits many of its passwords in cleartext—in particular, when a new user is created with the Perforce Visual Client. This condition was only discovered, however, in the default installation state of security level zero.

Finding P-4—System/user/workspace enumeration

Using the P4 command-line client, the P4V client, and the P4Web interface, an attacker can create a new user and browse Perforce servers, the users and groups on the system, the workspaces available, and many other features and settings of the server. The following commands allow the attacker to enumerate many of the Perforce settings, including the tickets issued to the user when logging in:

```
branches      Display list of branches
changes       Display list of pending and submitted changelists
changelists   Display list of pending and submitted changelists
clients       Display list of known clients
counter       Display, set, or delete a counter
counters      Display list of known counters
depots        Display list of depots
describe      Display a changelist description
diff          Display diff of client file with depot file
diff2         Display diff of two depot files
jobs          Display list of jobs
labels        Display list of labels
license       Update or display the license file
monitor       Display current running perforce process information
opened        Display list of files opened for pending changelist
protects      Display protections in place for a given user/path
sizes         Display size information for files in the depot
tickets       Display list of session tickets for this user
users         Display list of known users
workspaces    Display list of known clients
```

By allowing this somewhat anonymous level of enumeration, it provides the attacker with the building blocks for a directed, targeted attack.

Finding P-5—All communications between client and server are unencrypted

All information, including source code, communicated between the P4Web client and PerForce server and P4V client and PerForce server can be easily sniffed and compromised by a malicious user on the shared network. The absence of SSL undermines several aspects of application security. Consider the following vulnerabilities:

- Eavesdropping
  - » Using a packet sniffer, a malicious user could record sensitive data handled by the application
  - » Users' credentials could be sniffed, allowing the attacker to log in to the application and impersonate the victim
  - » The session identifier could be recorded, allowing the attacker to impersonate the victim by replaying this identifier when communicating with the server
- "Man in the Middle" attacks
  - » Without SSL, the user has no way of knowing whether the server is really the system it claims to be. Attacks such as address resolution protocol (ARP) poisoning, domain name system (DNS) poisoning, malicious proxies, and altered "host" files could be used to route a victim's traffic through an attacker's system before it reaches the server.
- Data corruption
  - » Because the traffic is not encrypted, the communicating parties are not authenticated (generally, secure sockets layer [SSL] provides authentication of the remote website to the user) and no integrity checking exists for each message being passed, so it would be possible for an attacker to alter data in transit. This would effectively lead to data corruption of the attacker's choosing.

Finding P-6—Authenticated users stay logged in

Once a user is logged in with a password, the user is issued a "ticket" that allows the user to stay logged in for the duration of the ticket which, by default, is 12 hours. The means an attacker can access the Perforce depot without logging in. This allows for a spoofing attack on the Perforce user base.

Finding P-7—Authentication for P4Web can be bypassed

During our review we identified multiple issues with its authentication. One of the highest risk vulnerabilities is that the tools do not enforce strong authentication. Any user can replay a request with a cookie value that is easy to guess and obtain authenticated access to the system. Once authenticated access is obtained, a user can perform very powerful operations on the PerForce server.

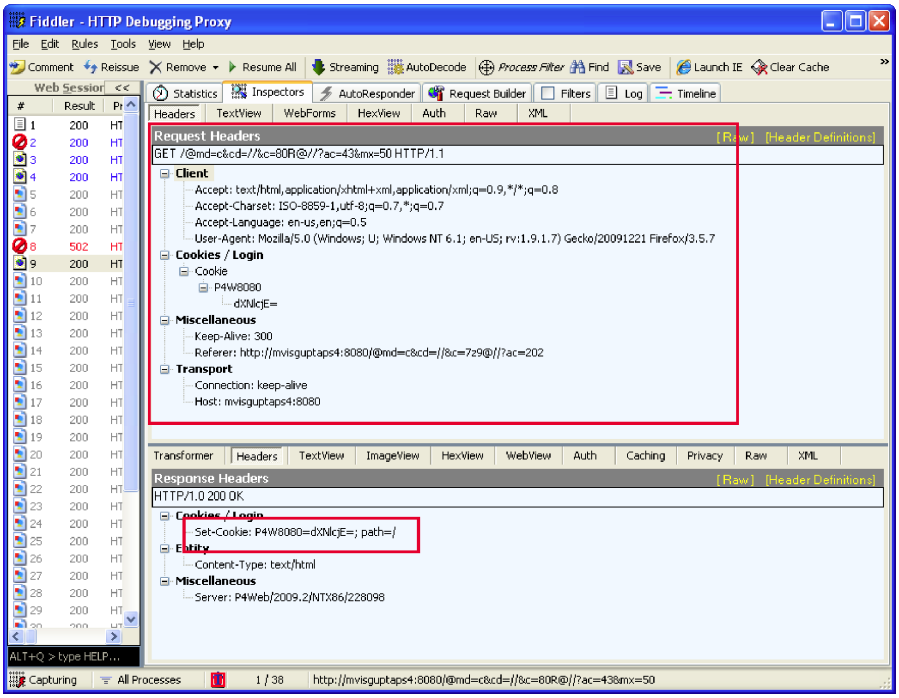


Figure 2. The session cookie is set to the base64 encoded version of the username.

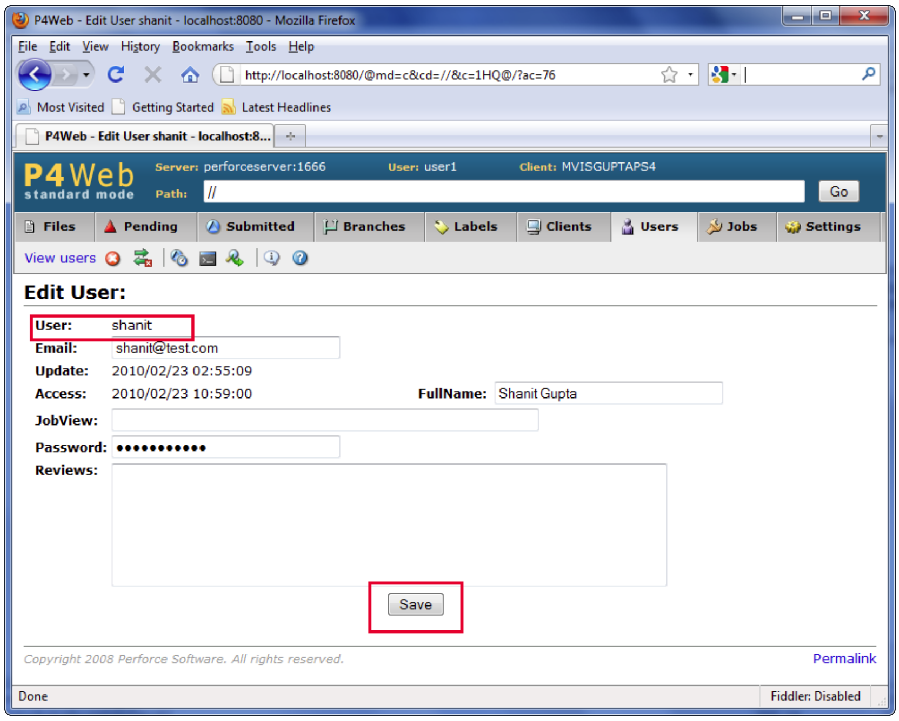
To obtain authenticated access, the attacker needs to replay a request like the one displayed in the screen shot above. As long as the attacker provides the right URL and sets a cookie with the name of the victim's account, he will be granted authenticated access to the system. It is not difficult to determine the name of the victim. This information is easily accessible from the P4Web application.

Finding P-8—Multiple authorization failures

During our review, we observed multiple lapses in authorization controls. The P4Web web interface hides controls that should not be available to the logged-on user. However, by manipulating the URL and the parameters passed, a malicious user can access and edit other user/client/project information.

The risk of this vulnerability is increased when combined with prior findings. After gaining authenticated access, the malicious user can exploit the authorization failures and expand his influence.

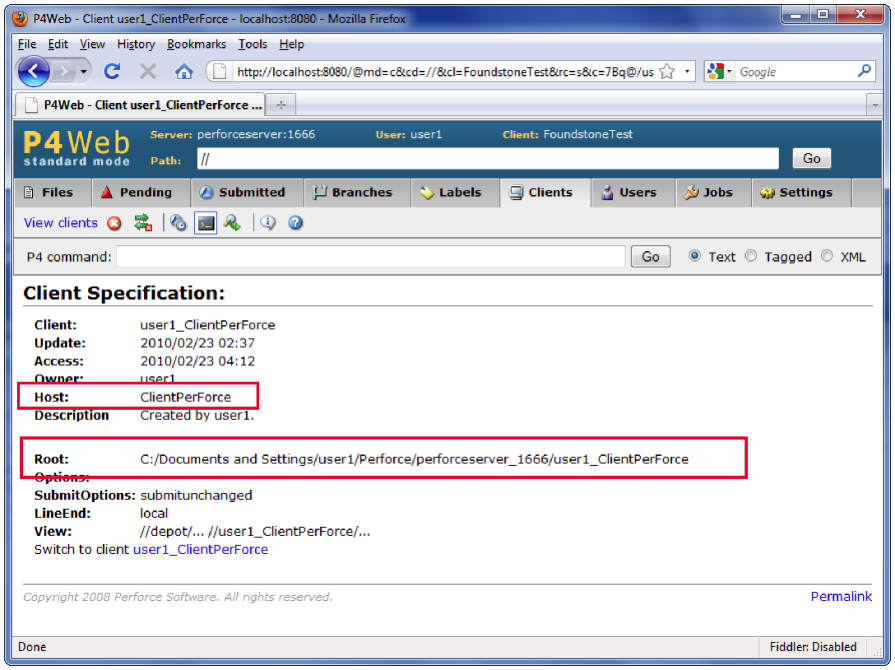
For example, the logged-on user in the screen shot on the next page, "user1," should not have access to the "shanit" user. But by manipulating the URL and the parameters passed, we were able to edit the profile page of user "shanit".



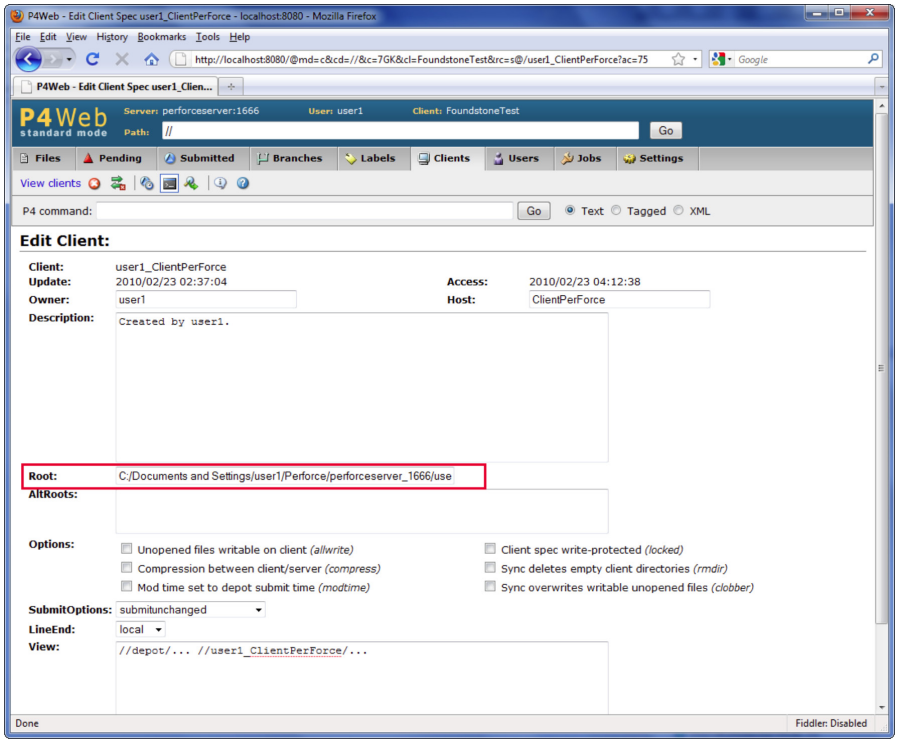
Finding P-9—Directory traversal attacks can lead to system compromise

Authorization failures combined with inadequate access controls on user workspaces can allow a malicious user to perform directory traversal and compromise all files on a victim's system. Furthermore, a malicious user could potentially overwrite files or compromise the Windows security accounts manager (SAM) information to gain complete access to the victim's system.

All users of PerForce SCM solution maintain a workspace on their local systems. This workspace is the local cache of the files that are saved at the SCM. The location of the workspace is configurable by the end user. A malicious user can exploit the weak authorization controls and change the workspace location for other users. The workspace's root directory can be changed to "C:\Windows" or any other sensitive location of the attacker's choice. The next time a "sync" operation is performed (explicitly or implicitly), the files from the workspace could get written to PerForce server and get shared with other users. Files may also get overwritten on victim's workspace, thereby overwriting key executables and configuration which can lead to system compromise.



The logged-on user should only have read access to client "user1_ClientPerForce". This is indicated by the lack of an "edit" button for this client.

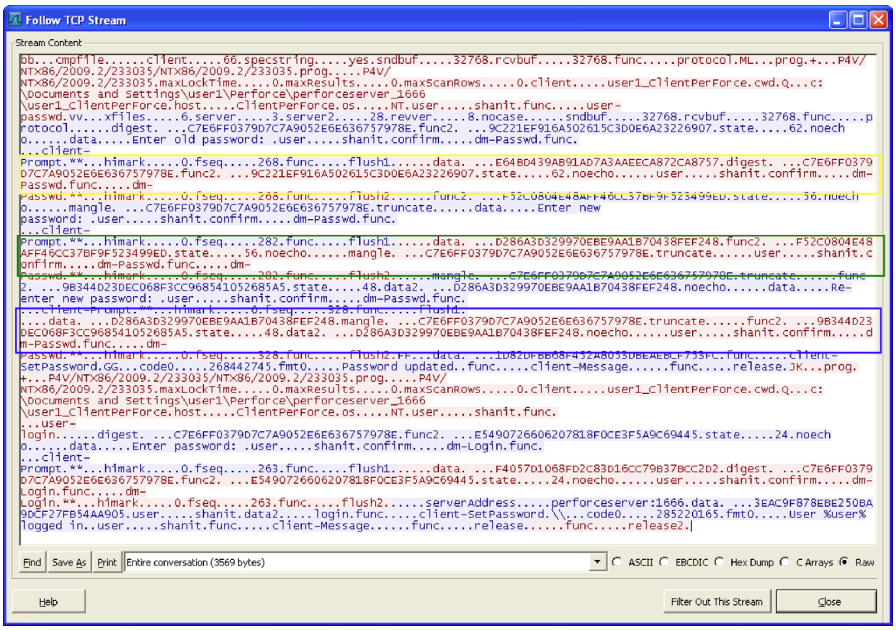


By manipulating the URL and the parameters, the logged-on user can change the root directory on the victim's system, thereby pointing to a more sensitive location.

Finding P-10—Unauthorized password change may be possible

After successful authentication between the P4V client and the PerForce server, a session token is established. This session token may be used to identify a user session. However, the session token does not seem to be used for change password feature. This allows a malicious user to perform an unauthorized change of password.

To perform this attack, the malicious user would need to initiate a change password session. For the first request, the attacker would need to send his username and current password's digest. In the request that contains the new password's digest and existing username, the attacker could send the victim's username and a password digest selected by the attacker. The same would be repeated in the password confirmation request. This would allow an attacker to change password belonging to victims.



The section highlighted in yellow displays the request sent with the logged-on user's username and password digest. The section highlighted in green is the request with the new password digest. This request also contains the username. A malicious user can change this username to the victim's name. Similarly, the request highlighted in blue is the password confirmation. Once again, the malicious user can change the username to the victim's and potentially change the victim's password.

Finding P-11—Journal and Log files are world readable

By default, Perforce installs its files, including its journal and log files as world readable. This allows an attacker to view all entries in the journal and log that include all the database updates and source code commits even as a standard user.

Finding P-12—PerForce files stored in clear text

The Perforce P4V client and the PerForce server store all files in cleartext on the client and server system.

If a developer's system were to get compromised, this could result in compromise of the all the code on the local cache. Similarly, cleartext storage of all files on the PerForce system makes code an easy target in the case of a system compromise.

Finding P-13—By default, accounts have high privileges

By default, all users are granted super-user access, and any user is allowed to create an account on PerForce server. The first user to execute "p4 protect" becomes the super user of the environment.

While this instruction is documented, it may not be very obvious to the administrator that the environment needs to be locked down before it is made open in a distrusted environment.

Finding P-14—Socket hijacking could potentially lead to information disclosure and/or denial of service attacks

An insecurely bound socket can be hijacked by an attacker, allowing the attacker to launch information disclosure and/or denial of service attacks.

The way the attacker accomplishes this is by binding the listener socket to the same TCP/UDP port (as the vulnerable socket) and a specific IP address (one of the underlying IP addresses). When the client makes a connection request to the server, the attacker's listener socket receives the connection (instead of the real socket) since it is listening on a specific IP address. The client is unaware that it has connected to a rogue listener socket, allowing an attacker to launch successful information disclosure attacks. Additionally, since the real socket never receives the client connection, it also allows the attacker to launch successful denial of service attacks against the server.

To successfully exploit this vulnerability, the attacker needs to have minimum access to the system that hosts the vulnerable socket and allows the attacker to execute code/applications on the system.

Countermeasures

The following recommendations can help protect sensitive servers from attack. All possible perpetrators and all vectors should be considered when installing and enforcing a secure SCM system. Potential perpetrators include;

1. Insiders (non-malicious) with privileges
2. Insiders (malicious) with privileges
3. Outsiders (non-malicious) with privileges
4. Outsiders (malicious) with privileges

Below are our high-level findings and recommendations:

- *Confidentiality and integrity*—User accounts typically fail to follow the "least privilege" principle of security, meaning they are often granted excessive authorization to functions within the system as well as to the actual source code files themselves. Proper procedure should limit access to administrative functions as well as limit read and/or write access to source code and source code trees. Once an attacker has raised his access to "administrative" level, he has complete control of the SCM. There are rarely any restrictions to the administrative level of access. It is vital that organizations enforce a "need-to-know" access control model while providing all access to source code control systems.
- *Logging, auditing, and non-repudiation*—Little log auditing is configured by default on SCM systems. Instead, the system must be explicitly set up and configured to log each event performed on the system, including source code checkouts, commits, branches, and configuration changes and should include the user, time/date, and function performed. This is especially true for significant functions such as attempts to download the entire tree and failed attempts to access sensitive functions and areas. The logs must also be set up to be backed up and stored at an offsite facility to ensure their integrity. If possible, logging directly onto a system other than the SCM itself is preferable. Finally, it is imperative that these logs be audited on a regular basis.
- *Availability*—The source code management system should be resistant to denial-of-service attacks both from the network, the system, and the SCM application itself
- *Authentication*—The system should employ 2FA (two-factor authentication) to ensure that users logging into the system are who they say they are. Where available, the SCM should be 2FA aware and leverage the two-factor nature of authentication to tie into its authorization framework. One-time passwords (OTP) should also be considered as an alternative to 2FA. Additionally, all user and administrative accounts should be audited to remove all retired or removed users from the system.
- *Self-protection*—The system must be able to protect its logs and configuration files to block attempts to subvert. Additionally, the algorithms in use to perform that self-protection must be strong. Cryptographic

hashing can provide a strong basis for determining if files change in an unauthorized way.

- *Backup*—Regular backups should be performed including all source code. Backup integrity should be validated independently and all files digitally signed
- *Communications*—The system encrypts data at rest, data in motion, and data in use. This means that all communication, whether the traffic is created by a user logging onto the system, accessing a specific branch, or performing any other operation, must be encrypted. Additionally, the repository and all backups must be encrypted.
- *Patching and configuration management*—The operating system housing the SCM software must be patched on a regular basis and its configuration continually monitored and updated based on the latest threats. The same is also true for the components of the SCM system itself.
- *Overall system hardening*—The SCM should be housed on a single-use system. It should not serve other purposes beyond SCM. The underlying operating system should disable all non-essential functionality. Prior to deployment of the SCM, administrators should implement a good common hardening framework such as FISMA or the National Security Agency (NSA) hardening guidelines.
- *Network logging*—Due to the nature of system hacking (being able to make any changes on a system that has been compromised), one of the only mechanisms for accurately understanding how an attacker gained access onto a system (and then defend against future attacks) is to trace the attacker's steps through network logging. Most companies have inadequate network infrastructure logging including ingress/egress firewall logging, VPN logging, email/IM logging, web logging, network, and router logging. Additionally, very few organizations have network forensics systems that store all traffic for offline analysis after an event. This countermeasure can be critical to understanding what truly happened in attacks such as Aurora.

Note: It is important to note that each of these findings applies to all of the interfaces a particular SCM system exposes. In our experience, it is often the less-used interfaces, such as the web interface or command line interface, that get the most attention from attackers, so it is important that these are not ignored from a defense-in-depth perspective.

Perforce-Specific Recommendations

Beyond the general security recommendations and guidelines above, the Perforce-specific recommendations can be found below. Additionally, Perforce security recommendations are available at http://kb.perforce.com/article/1173/basics-of-perforce-security and at http://www.perforce.com/perforce/conferences/us/2007/presentations/DSteele_Authentication2007.pdf.

Finding P-0—Perforce performs little to no security hardening of the system it installs on

Countermeasure—Harden the underlying system.

Consequently, companies should practice general security hardening of all systems with source code installed and hardening of Perforce servers according to corporate security baselines at a minimum. Many security hardening benchmarks and checklists are readily available from the Center for Internet Security (www.cisecurity.org) and NIST (http://web.nvd.nist.gov/view/ncp/repository).

Finding P-1—Perforce server service (p4s.exe) and Perforce web (p4webs.exe) installs with SYSTEM level privileges

Countermeasure—Change the user that p4s.exe and p4webs.exe runs as to a least privileged user

Based on your company's needs, set up a Perforce user that runs with the fewest privileges possible. Modify the running user by selecting "Control Panel->Administrative Tools->Services," selecting "Properties" for the running Perforce service (Perforce and Perforce Web), and selecting the "Log On" tab. Select a user with the fewest privileges possible to run the Perforce applications.

Finding P-2—Unauthenticated user creation

Finding P-4—System/user/workspace enumeration

Finding P-7—Authentication for P4Web can be bypassed

Finding P-13—By default, accounts have high privileges
Countermeasure for Findings P-2, P-4, P-7, P-13—Enable "Administration" or "p4 protect" access.

By default, anyone can gain access to existing source code within Perforce. As a result, you must enable "Administration" or perform the "p4 protect" operation within Perforce to restrict access to existing users and control access to the source code. From here, you can restrict users' authentication and authorization to sensitive information in the depot. Be sure to enable the highest security level of three.

Finding P-3—Many passwords are unencrypted
Countermeasure—Employ some form of network or end-to-end encryption such as Layer2 or Layer3 SSL.

Finding P-5—All communications between client and server are unencrypted
Countermeasure—Encrypt all traffic using SSL or SSH.

Best practices for configuring SSL security include:

- Deploy a unique SSL certificate generated by a trusted certificate authority for each system (or a wildcard certificate that achieves the same result)
- Disable support for the SSLv2 protocol, which has known "Man in the Middle" and encryption downgrade attacks
- Disable support for all cipher suites with encryption keys less than 128 bits
- Disable support for all cipher suites that support "Anonymous Diffie-Hellman" for key exchange
- Require SSL on the login form and every page that requires authentication to access

Proper SSL configuration may be tested using the SSLDigger tool by McAfee Foundstone®. SSLDigger may be downloaded at http://www.foundstone.com/us/resources/proddesc/sitedigger.htm.

Finding P-6—Authenticated users stay logged in
Countermeasure—Set expiration through the "TIMEOUT" option.

Finding P-8—Multiple authorization failures
Countermeasure—Disable p4web access.

Finding P-9—Directory traversal attacks can lead to system compromise
Countermeasure—Disable p4web access.

Finding P-10—Unauthorized password change may be possible
Countermeasure—The fix is due in the 2010.1 release.

Finding P-11—Journal and Log files are world readable

Finding P-12—PerForce files stored in clear text
Countermeasure for Finding P-11 and P-12—Protect your system against attack to prevent an attacker from gaining user or administrative level access to the system directly.

Finding P-14—Socket hijacking could potentially lead to information disclosure and/or denial of service attacks
Countermeasure—McAfee recommends securely creating the listener sockets by turning on the exclusive address use option (SO_EXCLUSIVEADDRUSE) on the socket so that an attacker cannot hijack the server socket. Additionally, the process can also set an access control list on the socket if the underlying operating system is Windows Server 2003 or above.

McAfee Protection

Customers can deploy a number of McAfee products to help protect SCM and content management systems from attack. The following technologies from McAfee can help secure your systems from similar attacks in the future:

- *McAfee Vulnerability Manager software*—Using discovery and vulnerability checks to find SCM systems on your network as well as the vulnerabilities present in the system, McAfee Vulnerability Manager software detects many of the security weaknesses in systems that have been compromised. See www.mcafee.com/us/enterprise/products/risk_and_compliance for more information.
- *McAfee Policy Auditor software*—Using configuration audit checks to determine the most secure configuration of a system, McAfee Policy Auditor software detects the security weaknesses in the systems that have been compromised. See www.mcafee.com/us/enterprise/products/risk_and_compliance for more information.
- *McAfee Endpoint Encryption software*—Deploying McAfee Endpoint Encryption software reduces the impact of the attack by restricting access to the core assets and requires significant additional work for the attackers to bypass. See www.mcafee.com/us/enterprise/products/system_security for more information.
- *McAfee Data Loss Protection (DLP) solutions*—Deploying McAfee Network and/or Host DLP solutions allows you to prevent and detect the extraction of sensitive information such as source code from outside the company. See www.mcafee.com/us/enterprise/products/data_protection for more information.
- *McAfee Configuration Control software*—McAfee Configuration Control software allows you to disallow any configuration changes to the system, protecting your SCM systems from being modified to extract sensitive information. See www.mcafee.com/us/enterprise/products/risk_and_compliance for more information.

Conclusions

The use of APTs is on the rise by a growing group of malicious attackers committed to their targets. The targets have now moved beyond the defense industrial base (DIB), government, and military computers to include corporate and global commercial targets. More and more, these attacks focus not on using and abusing machines within the organizations being compromised, but on the theft of specific data and intellectual property. It is therefore vital that organizations work proactively toward protecting the heart of their value: intellectual property. Enterprises need to take action to discover these assets in their environments, assess them for vulnerabilities and misconfigurations, and protect them from misuse and attack.

Credits and Acknowledgements

This white paper was a collaborative effort among numerous McAfee Foundstone Professional Services consultants, McAfee employees, executives, and researchers. Significant contributors include Stuart McClure, Shanit Gupta, Carric Dooley, Vitaly Zaytsev, Xiao Bo Chen, Kris Kaspersky, Michael Spohn, and Ryan Permeh.

We would also like to thank the Perforce team for their assistance in validating our findings and working with us to provide remediations and countermeasures.

About McAfee Labs

McAfee Labs delivers the core technologies and threat intelligence that power the McAfee suite of endpoint, web, email, and network security products. With a research footprint that covers the globe, McAfee Labs provides accurate and predictive global threat intelligence. A team of 350 multidisciplinary researchers in 30 countries follows the complete range of threats in real time, identifying application vulnerabilities, analyzing and correlating risks, and enabling instant remediation. Support from McAfee Labs' 24/7 emergency response team ensures the highest level of insight into emerging risks.

About McAfee Foundstone Professional Services

McAfee Foundstone is the pre-eminent leader in network security consulting, serving hundreds of high profile organizations in the Fortune 500, federal and state governments, and the military. Our bonded consultants are recognized experts, educators, and technical authors who have helped protect the networks of many of the world's most at-risk organizations.

About Perforce

Founded in 1995, Perforce Software develops, markets, and supports Perforce, the Fast Software Configuration Management (SCM) System. Perforce SCM versions and manages source code and digital assets for enterprises large and small.

Perforce Software is a privately held company headquartered in Alameda, California with international offices in Wokingham, United Kingdom and Sydney, Australia.

Today more than 320,000 developers at 5,000 organizations worldwide use the Perforce SCM System to manage their source code and digital assets.

McAfee, Inc.
3965 Freedom Circle
Santa Clara, CA 95054
888 847 8766
www.mcafee.com

8615wp_OperationAurora_0310