



Hired to hack an ERP System

October 24, 2006

By John Verry

Contact John 

John Verry is the Principal Enterprise Security Consultant for Pivot Point Security, which specializes in internal controls auditing, ethical hacking, and security event management. He has conducted security assessments for numerous Fortune 100 companies, healthcare providers, state and local government agencies, and not-for-profit organizations. Over his 15 year career in information technology he has core experience in application development, relational database management systems, network architecture/security, and information systems auditing. He is a Certified Information Systems Auditor (CISA) and a Checkpoint Certified Systems Expert (CCSE).

Takeaway:

A major governmental entity called on Pivot Point Security to certify the security posture of a critical ERP application that handles sensitive employee data prior to deployment. Pivot Point delivers with a report that illustrates that a holistic view and execution of IT security is needed to best protect that critical data.

Based on previous work, our name, Pivot Point Security, came up when a large government agency sought an independent third party to conduct a security assessment of a large scale (100,000 + total end users) ERP (Enterprise Resource Planning solution) rollout.

Working in concert with the PM/QA vendor we were engaged to provide assurance to the Agencies Executive Management Team that the project would achieve critical security objectives. In light of the recent spate of head line grabbing events in the governmental sector we were not surprised to learn that many of their objectives revolved around the prevention of Identity Theft. Accordingly our efforts focused on providing assurance that their ERP rollout appropriately ensured the confidentiality of the sensitive employee data contained therein (e.g., payroll, social security numbers).

The engagement consisted of a comprehensive security assessment that included a vulnerability assessment and penetration test of the systems and subnet hosting the application, the network that the application was administered from, and the application itself. We also performed a controls audit for those controls that were deemed critical to achieving the confidentiality objectives.

Off to the races

After running a Vulnerability Assessment against the subnet the application resided on, we sat down as a team to review the data and formulate our Penetration Testing plan. Typically we look to the Vulnerability Assessment data to help us identify which combinations of vulnerabilities represent the most probable path to achieving the Penetration Testing objective--in this case, accessing confidential data. The subnet and systems appeared to be very well secured. We prioritized the web servers and several network devices as they appeared to have a few

potentially vulnerable services running. However, we were not optimistic that we would be able to directly "hack" the application/data.

Fundamentally, where direct access is not possible, the goal is to garner access to a system and then systematically escalate privilege level so that we can control the system by gaining administrative privileges. This provides us with a wealth of options including:

- The ability to extract encrypted passwords from the registry.
- Access to sensitive data or applications.
- A springboard system we can use to attack other systems. Often the technical controls in place are lower for systems which are "trusted."
- The ability to modify the system to secure future access via backdoors.
- The ability to plant keystroke loggers to covertly monitor a user's activities.

These options provide the ability to escalate privilege level until achieving administrative level on a critical system and/or at a domain level.

Point-Counterpoint

The web server of interest was running Open SSL, which we thought may be vulnerable to a brute force password attack. The connection was refused and a quick re-scan of the box showed us that the once open port was now closed. We suspected that our activities were being monitored and an administrator was actively defending the server (despite management's directive not to). We examined the FTP and SMTP services running on this server and in doing so were able to determine that a user account "ERPadmin" existed, but our brief password guessing foray was unsuccessful as we were locked out after five failed attempts. We determined the existence of the account by differentiating the responses for this account and a known non-existent account (e.g., PivotPoint!). Account lockout being in place, in concert with the low levels of technical vulnerability, enforced our opinion that the subnet was very well secured.

As the network devices of potential interest supported other production applications we delayed evaluating these devices and instead moved onto the organization's LAN.

Almost only counts in Horseshoes

At the onset of the engagement we encountered a significant amount of resistance to our suggestion that we include the LAN in the security assessment. Our belief (fostered by numerous engagements of a similar nature) is that those LAN segments that may have privileged access or where users of a privileged nature may reside are a significant source of risk, even when the application is on a different subnet.

Our vulnerability scan on the LAN was almost as clean as the production subnet. It was a nearly homogeneous Windows network (2000 Servers and XP hosts) with virtually all hosts identically configured. The "nearly" became a problem when we realized there was an orphan NT4 server running an older custom application on the LAN. Password Complexity settings were centrally distributed by Windows SMS, which unfortunately does not work with NT4. This resulted in the use of same user name/password (jimmy/jimmy) account which we picked up with a Net bios enumeration tool (NBTEnum) that looks for common password weaknesses. When we realized the jimmy account was a local admin account we realized we had our first foothold.

Moving up and on

Once local Admin privileges have been garnered on a system it is possible to use a tool (e.g., PWDump) to retrieve the SAM from the machine. The SAM is an encrypted list of passwords. We fed the same into a password cracking application (e.g., L0phtCrack 5) and executed a dictionary attack. The dictionary attack cracked two additional user accounts but neither provided us with greater privilege than we already possessed. There was one additional account that we suspected was a common LAN administrative account based on its name. We kicked off a brute force attack against this account and stepped away to grab a cup of coffee while the cracker did its work.

On returning, we sought and received permission to conduct our Penetration Testing activities against a network switch of interest. It was running an older version of Cisco's IOS that we thought may be vulnerable to an ARP Spoof. When successful, an ARP spoof "tricks" the switch into believing that your attack machine is the destination for all traffic on the LAN segment (Go [here](#) for a more detailed definition). Essentially, this resulted in our attack box acting as a switch, with all network traffic flowing through our machine.

Observing the network data we noticed that all users on the network were authenticating using NTLM (the default encryption for Windows NT and 2000). We sent modified packets back to the end user's workstations letting them know "we don't understand NTLM, please send unencrypted passwords." This trick didn't work, however; "Please send us passwords using LM authentication" did. LM uses a weaker variety of encrypted authentication which can potentially be broken. We gathered these encrypted passwords for future cracking. Slowly and methodically we were increasing our avenues for privilege escalation and were growing confident that our activities would be rewarded.

Cheerios and local admin privileges

Breakfast tasted a little sweeter on noting that during the night the password cracker had cracked the other Local Admin privilege which proved to be a common password used to administer all computers on the LAN. This provided the avenue we needed to garner admin level privilege to the ERP system. But first we needed to determine whether there was a keystroke logger that our client's antivirus solution (in this case one of the most prominent enterprise security companies in the industry) wouldn't catch.

Based on our research on the AV vendor's site, multiple keystroke logger sites, and several hacker forums we selected a non-widely publicized highly covert keystroke logger. We connected to our target workstation via RDP and, with fingers crossed, loaded the logger onto a non-technical person's machine whom we hoped we could sidetrack if it threw a warning message. When the keystroke logger started reporting back the user's activity--we knew we were one last step away. Waiting for lunchtime (so as to avoid detection) we then loaded the same keystroke logger on four additional workstations; those belonging to the ERP Administrator, a DBA, and an AIX administrator. Within an hour each user had logged on to administer their element of the solution and the keystroke logger reported the passwords back to us. Logging in to the systems with our new found admin level privileges, we had fully compromised the ERP application, all of its systems, and the data contained therein. We had complete and unfettered access to all of the highly sensitive information of the client's personnel contained in the database including names, addresses, social security numbers, payroll information--a wealth of information that could be leveraged by an individual with malicious intent in a myriad of ways.

Caught red handed

To test the powers of observance of several of the individuals whose accounts we had compromised, we left ourselves logged into their machines with our Local Administrator privileges. We were impressed by how quickly our actions were detected including the presence of the keystroke logger. The admin was knowledgeable enough to run an application to garner a better view of all processes that were running in combination with an application to monitor network connections (e.g., Vision and netstat). When he noted his machine connecting to our IP, we were nabbed in the act.

This real world exercise in ERP security illustrates how the significant efforts made to directly secure the application, database, and local subnet, can be subverted by a seemingly small breakdown in the control environment of the organizations LAN. The strong set of technical controls (in this case the overall high level of security of the ERP application) were relatively quickly undermined by a weak set of internal controls, in this case the failure to enforce a strong password policy on the "jimmy" account which set off a cascade of escalations, that ultimately led to the potential exposure of critical company data. Our ARP spoofing efforts would also have resulted in a total system compromise had we continued our efforts.

It is important to note that we also conducted application level penetration testing and a fairly in depth audit of critical controls in the areas of data segregation, data ownership, authentication, authorization, and change management. These efforts provided management with a higher level of assurance that key security objectives were being achieved.

Following the delivery of our findings to management, the governmental agency took the legacy NT box with the "jimmy" account off line, effectively enforcing the security policy across a homogeneous Windows 2000 environment, updated all their desktops to the latest antivirus engines and definitions in order to detect keystroke loggers, and updated the IOS on all of their Cisco devices that were vulnerable to ARP spoofing.

As a result of the formal report of our methodologies and findings, management of the organization obtained assurance that the risk of exposure of sensitive employee data relating to the deployment of the ERP application had been mitigated to an acceptably low level.

Noting the value of the exercise, and moving towards best practice, the client has scheduled security review activities into all future project phases.

John Verry is the Principal Enterprise Security Consultant for Pivot Point Security

White Papers

- [Improving Vulnerability Management with Penetration Testing](#)
- [On-demand Webcast: Combating the Elevated Threat of Spyware in Today's SMBs](#)
- [The SMART Way to Secure Messaging for Microsoft Exchange Environments](#)
- [Business Resilience: Proactive Measures for Forward-Looking Enterprises](#)
- [Upgrade to Next Generation Antispam/Antivirus for Exchange: Download Ninja Today! \(Trial Download\)](#)

Additional Resources

[Tech Toolshed](#)

<http://www.techtoolshed.com>

[Learning CD-Roms](#)

<http://fasttrack.techrepublic.com>

[Quick Reference Charts](#)

<http://quickref.techrepublic.com>

[TechRepublic's Catalog](#)

<http://www.techrepublic.com/catalog>

[Copyright](#) ©1995- 2006 CNET Networks, Inc. All Rights Reserved.
Visit us at www.TechRepublic.com