Samantha Ho

Professor So

ITP 125

9:30-10:50 am

**Final Paper**

**Black Hat 2013: Exploiting Network Surveillance Cameras Like a Hollywood Hacker**

Craig Heffner introduces the topic of exploiting network surveillance cameras like a Hollywood hacker. He demonstrates vulnerabilities in web servers running through surveillance cameras, with D-Link, Cisco, and IQ products as examples. All of these cameras provide an administrative interface as well as access to the video feed through a web server running through the camera, which makes the web server a very attractive target for an attacker.

In his first example, Heffner explores D-Link surveillance cameras and their web server. D-Link uses lighttpd, which is an open source web server imbedded in many devices. He finds that cgi-bin is unrestricted. Cgi-bin only has one file, rtpd.cgi. Rtpd.cgi is a shell script that can be used to start and stop the real time transport protocol daemon. The program will execute whatever is in the query string and run as root. It sends a response back to browser. He has an example in which he enters a command in the query string and the program sent the administrator credentials to his browser. He is now both root and administrator. Since D-Link reuses code for most, if not all, of their products, he has access to most, if not all, of those products. Furthermore, some other lesser brands use the same code, making their products accessible as well.

In his second example, Heffner tries to find vulnerabilities in a camera from a more reputable vendor, Cisco. Cisco has a web server and enforces authentication by using .htpasswd protection in every directory that Cisco wants to be password protected. He finds that there is an unprotected file, oamp.cgi. Oamp.cgi implements its own mini API and is completely separate from everything else on the web server. Before executing an action, the program checks to see if the user has entered a valid session ID. Since he does not have a valid session ID, he must enter a username and a password. He finds that the only place in the interface that the username and password is referenced is in the configuration file. These values are hard coded into the device running config under the oamp section of the configuration file. The oamp interface and hard coded accounts are undocumented. No one knows they are there unless they bother to look at the firmware. The problem with hard coded secret passwords in the system and backdoors is that they cannot stay secret. The user can use the backdoor accounts to exercise the login action. This gives the user a session ID. This allows the user to run oamp.cgi actions, which includes downloading the configuration file. This gives the user administrator credentials.

In his third example, Heffner tries to find vulnerabilities in a camera from a brand that specializes in surveillance cameras, IQinVision. Once again, he finds a file he can get into without authentication, oidtable.cgi. Using the code behind oidtable.cgi, he is able to inject a command. He is able to retrieve arbitrary files, targeting the private password file, which contains the administrator credentials. The user can either decrypt the administrator password or overwrite the file.

In his fourth example, Heffner tries to find vulnerabilities in the firmware of the most expensive camera he found, from 3S Vision. He first encounters a restricted firmware download popup box in which the user has to enter a password. When he examines the java script code behind the popup box, he finds the password in the code, which does not bode well for the system’s level of security. He then sees that 3S Vision uses a custom web server, httpd. He examines how this web server handles authentication. He knows that the cameras use basic http authentication, which means that there will be base 64 decoding. He starts looking through the code for cross-references to base 64 decode. When the server decodes the password, it passes the password to base 64 decode, and conducts two string comparisons against a hard code of string to another hard code of string. Hard coding anything into http server (or any other server) is not a good idea, and he is able to easily gain administrator credentials.

In all of these examples, Heffner sifts through the code behind the firmware of the products to find any vulnerabilities he can exploit. The main point he is trying to get across is that anyone can gain access into a target’s network as root and/or administrator, if he or she is willing to look through thousands of lines of code for possible access points. Most, if not all, of the cameras reveal their model number in the login prompt even if the user is not authenticated. If an attacker knows the model number, he or she can find and analyze the firmware associated with the model for vulnerabilities and possible exploitations because all exploits are developed exclusively from firmware update files.

**Black Hat 2013: The Factoring Dead: Preparing for Cyptopocalypse**

There is a significant disconnect between theory and reality in security. There is lots of great, continuous academic research in cryptography, but few engineers get beyond *Applied Cryptography* before shipping code. There have been numerous attacks on the current TLS infrastructure. These attacks were not unpredictable if people had been abreast of the literature, a few of which predicted these attacks years ago. The disconnect results from a few factors. The first is that most systems are not designed for cryptographic agility, making hard to update the cryptosystems. The second is that cryptography is an ecosystem, meaning that the components involved have to be able to use the same algorithms and speak the same language. The third is that there are few companies that employ full-time cryptographers to read academic literature and apply it to their systems. The fourth is that it is hard for information security practitioners to keep up-to-speed. The fifth is that there is lots of momentum in the professional consulting core.

The current cryptosystems depend on discrete logarithm and factoring, which have seen some major new developments in the past months. There have been huge leaps in solving the discrete logarithm problem that have parallels to the past. There needs to be a move to stronger cryptosystems that leverage more difficult mathematical problems, such as Elliptic Curve Cryptography. There is a small, but real chance that both RSA and non-Elliptic Curve Cryptography Diffie-Hellman will soon become unusable. However, the ecosystem currently cannot support a quick pivot to Elliptic Curve Cryptography. In the meantime, there needs to be more research into alternatives to Diffie-Hellman and RSA. There is too much reliance on Diffie-Hellman and RSA that very few people have taken it upon themselves to look for alternatives.

The main point of this presentation is that people should not be complacent with current processes. The current process has heavily relied on Diffie-Hellman and RSA since the 1970’s. There have been very few feasible alternatives since then, most of which are not compatible with any current systems. If on the off-chance that both RSA and non-Elliptic Curve Cryptography Diffie-Hellman become unusable, there are no readily available alternatives to mitigate the aftermath. The cryptography ecosystem can collapse because it cannot incorporate potentially viable alternatives, such as Elliptic Curve Cryptography, in time to bridge the gap. Researchers need to be more proactive in finding alternatives to prepare for worst-case scenarios in the event the current infrastructure collapses.