LAST MINUTE CYBERSECURITY

Security incident report

Section 1: Identify the network protocol involved in the incident
The network protocols involved in this event were DNS and HTTP. However the transfer of the Malware payload was done on HTTP Protocol 1.1. HTTP is an application layer protocol and determines how data interacts with devices. DNS is the domain name system and the DNS helps networks know how to interconnect and route traffic, specifically in the change from web address to IP.

Section 2: Document the incident
A hacker repeatedly attempted to access administrator passwords with a brute force (repeating common passwords) attack. After guessing through the common default passwords, the hacker edited our website to redirect to a fake website that downloads malicious code. Following that a number of different customers of our website, filled complaints letting us know that they were prompted to download a file for free recipes, and now their computers are running more slowly. While the webadmin reaches out to the hosting provide our cybersecurity team stepped in to get a better understanding of the issue. We created a sandbox environment to access the compromised website while running the PacketSniffer TCPDUMP. At 14:18 this afternoon a normal DNS request made to the DNS server fed an

user the IP address of our server that hosts Yummyrecipiesforme.com
Within a few minutes the user's pc had shifted the port they were using on their machine, sent a direct SYN request to port 80 on our server, the connection was then accepted by our server. After that the user called a dated HTTP protocol (HTTP 1.1), and initialized the upload of a malware item to the server that hosts yummyrecipiesforme.com as evidenced by the push flag.

Next, we see the source computer migrate ports once again, and do a call to DNS where the great recipes. It initializes the same SYN requests with the Greatrecipiesforme.com servers, where it then establishes an http 1.1 connection using the get call once again.

Finally we studied the executable that we were prompted to download, and found that it was redirecting our browsers to a different website entirely. The hacker can then leverage this HTTP connection for further attacks of many kinds.

We found that the hacker got in because a default password was being used, and no protections for brute force were enabled.

Section 3: Recommend one remediation for brute force attacks
I think that enforcing 2FA is a very valuable part of any OS hardening activity. This is primarily because it forces the authentication system to verify both something that the person knows, and prove that they have something that they own or are. This will provide minimal slow down in user access, but save us from weak passwords, or leaked passwords exposing the entire company's data. It would have prevented the access to the server, and the person tied to the administrator credentials would have had an alert that multiple attempts were being made on their credentials.