HBGary Email Viewer

----@hbgary.com

0

Original file:	----
click here to show this e-mail with HTML markup	
From:	----@----
To:	----@----
Date:	Tue, 24 Nov 2009 10:34:45 -0500
Subject:	English Shellcode
click here to show full headers	
Attachments:	This e-mail does not have any attachments.

Fantastic research here. This technique poses significant challenges for
current in-line detection techniques and forensic investigations. Great
read.

Abstract:

"History indicates that the security community commonly takes a
divide-and-conquer approach to battling malware
threats: identify the essential and inalienable components of an attack,
then develop detection and prevention techniques
that directly target one or more of the essential components. This
abstraction is evident in much of the literature
for buffer overflow attacks including, for instance, stack protection and
NOP sled detection. It comes as no surprise
then that we approach shellcode detection and prevention in a similar
fashion. However, the common belief that components
of polymorphic shellcode (e.g., the decoder) cannot reliably be hidden
suggests a more implicit and broader
assumption that continues to drive contemporary research: namely, that
valid and complete representations of shellcode
are fundamentally different in structure than benign payloads. While the
first tenet of this assumption is philosophically
undeniable (i.e., a string of bytes is either shellcode or it is not),
truth of the latter claim is less obvious if there exist
encoding techniques capable of producing shellcode with features nearly
indistinguishable from non-executable content.
In this paper, we challenge the assumption that shellcode must conform to
superficial and discernible representations.
Specifically, we demonstrate a technique for automatically producing
English Shellcode, transforming arbitrary shellcode
into a representation that is superficially similar to English prose. The
shellcode is completely self-contained
i.e., it does not require an external loader and executes as valid IA32
code and can typically be generated in under
an hour on commodity hardware. Our primary objective in this paper is to
promote discussion and stimulate new ideas
for thinking ahead about preventive measures for tackling evolutions in
code-injection attacks."

http://www.cs.jhu.edu/~sam/ccs243-mason.pdf

Regards,

_____

---- | ---- | ---- | Telephone:
+1 xxx xxx xxxx | Mobile: +1 xxx xxx xxxx | ----@----.com