Security: How is a spoofing attack avoided in AD with Bright?

How is a spoofing attack avoided in AD with Bright?

[Everyone I speak with about joining a node to AD indicates that there is a unique key that the node receives at the time of the join so AD knows later that nobody is spoofing. How does Bright preserve this unique key information across re-provisioning?]

Answering the initial part first, about how spoofing is avoided:

This is dealt with by the standard method of using the Kerberos keytab file.

Kerberos/AD login authentication works by requesting a TGT (ticket-granting ticket) from the Kerberos KDC and then decrypting it with a key formed from the password entered locally. If that decryption works, the login is considered successful, if there is no keytab.

The problem with this approach is that the attacker could be simultaneously spoofing the KDC reply with a TGT encrypted in the password the attacker chose. If the system gets that reply before the real reply, it will happily decrypt it with the attacker's password and then consider the authentication successful. This is fairly easy to do if one has a system on the same local network, given that Kerberos is a UDP protocol.

If there is a local keytab, the login process takes one more step: it asks the KDC for a service ticket for the principal stored in the local keytab, and then validates that ticket by decrypting it with the key in the keytab. The attacker's KDC has no knowledge of the private key of the keytab on the system, and therefore will fail this step. This means, of course, that the system keytab needs to be locked down to be accessible only by root, since anyone who can access the keytab can still successfully attack the system.

Preserving the keytab during reprovisioning:

The keytab file is machine-independent and can be generated on Windows and then transferred to the Linux systems. You could either exclude such files from provisioning, or store a copy of them in a secure location on the head node and transfer them back after provisioning with a finalize script. The latter option will obviously also work in the case that disks of the nodes are re-formatted.

The KB article at http://kb.brightcomputing.com/faq/index.php?action=artikel&id=159 discusses SASL/GSSAPI Binds for LDAP searches, as this method does not require Administrator privileges on Windows, only permissions to join computers to the domain.

Setting up AD to work with Linux certainly can be troublesome sometimes. There are more than 8 methods to achieve that in RHEL6, for example. However, methods that work with a regular Linux system are expected to work with Bright Cluster Manager.

Unique solution ID: #1211
Author: Frank Furter
Last update: 2014-08-11 12:48