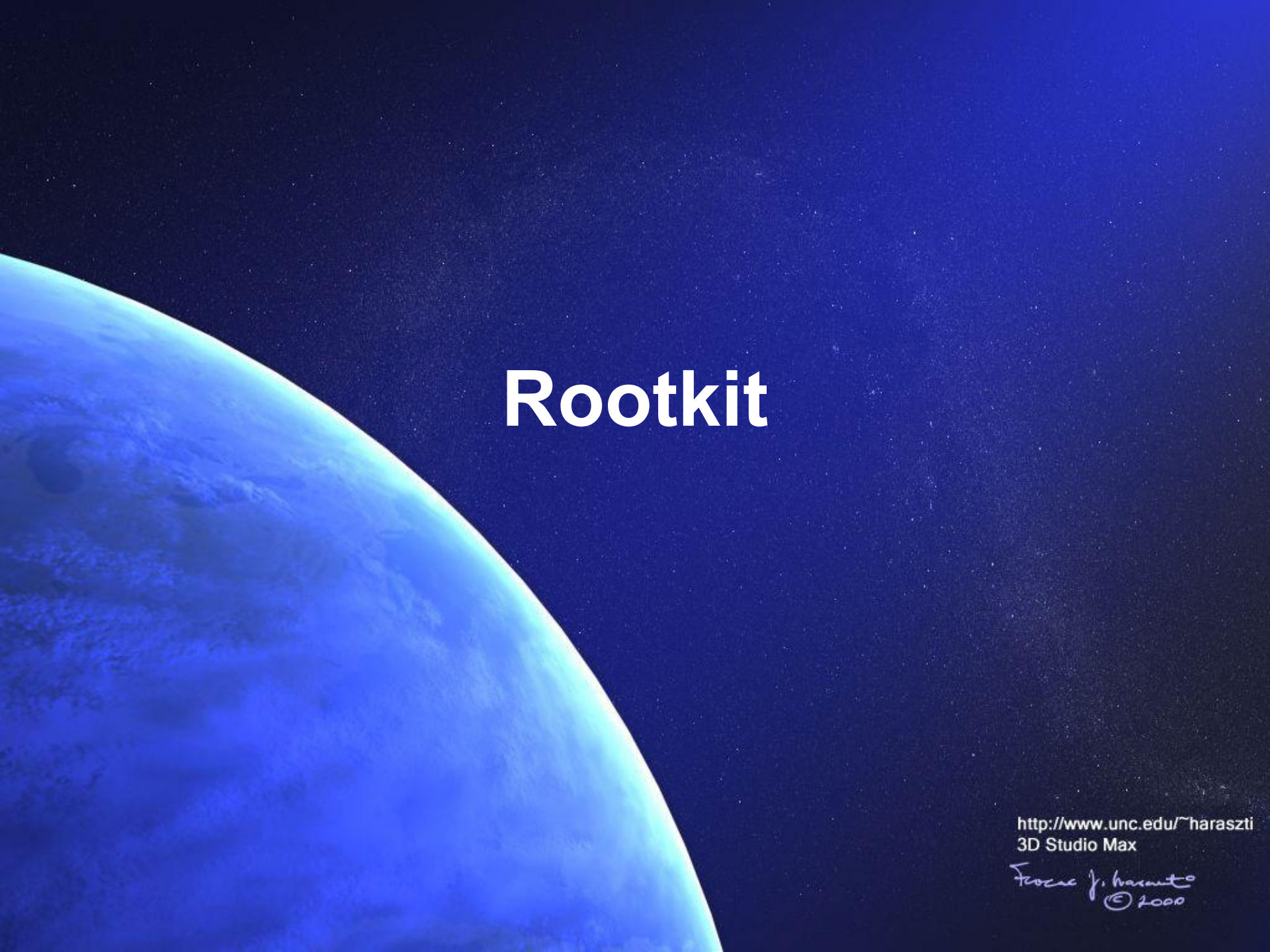
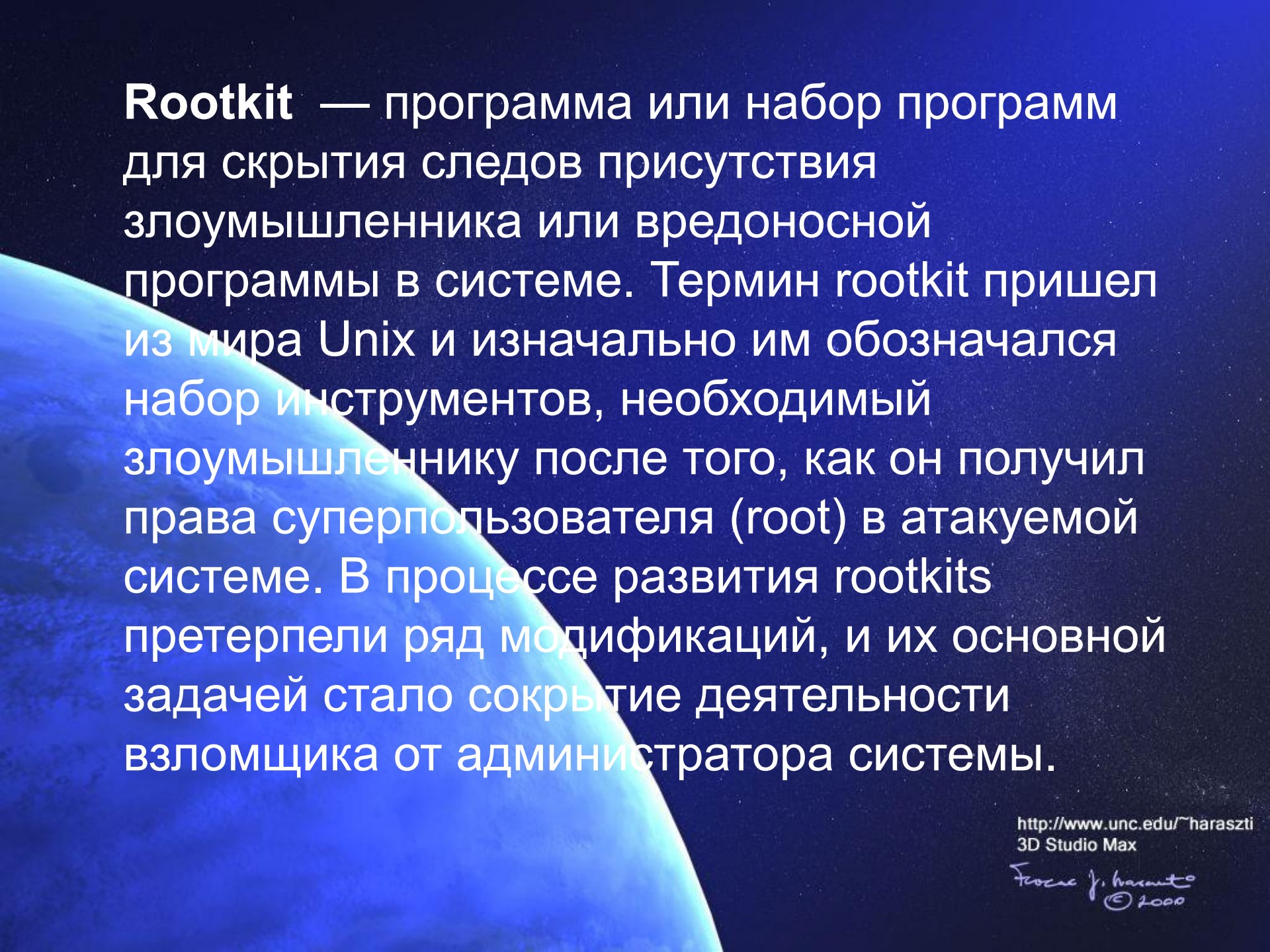
Rootkit



Rootkit — программа или набор программ для скрытия следов присутствия злоумышленника или вредоносной программы в системе. Термин rootkit пришел из мира Unix и изначально им обозначался набор инструментов, необходимый злоумышленнику после того, как он получил права суперпользователя (root) в атакуемой системе. В процессе развития rootkits претерпели ряд модификаций, и их основной задачей стало сокрытие деятельности взломщика от администратора системы.



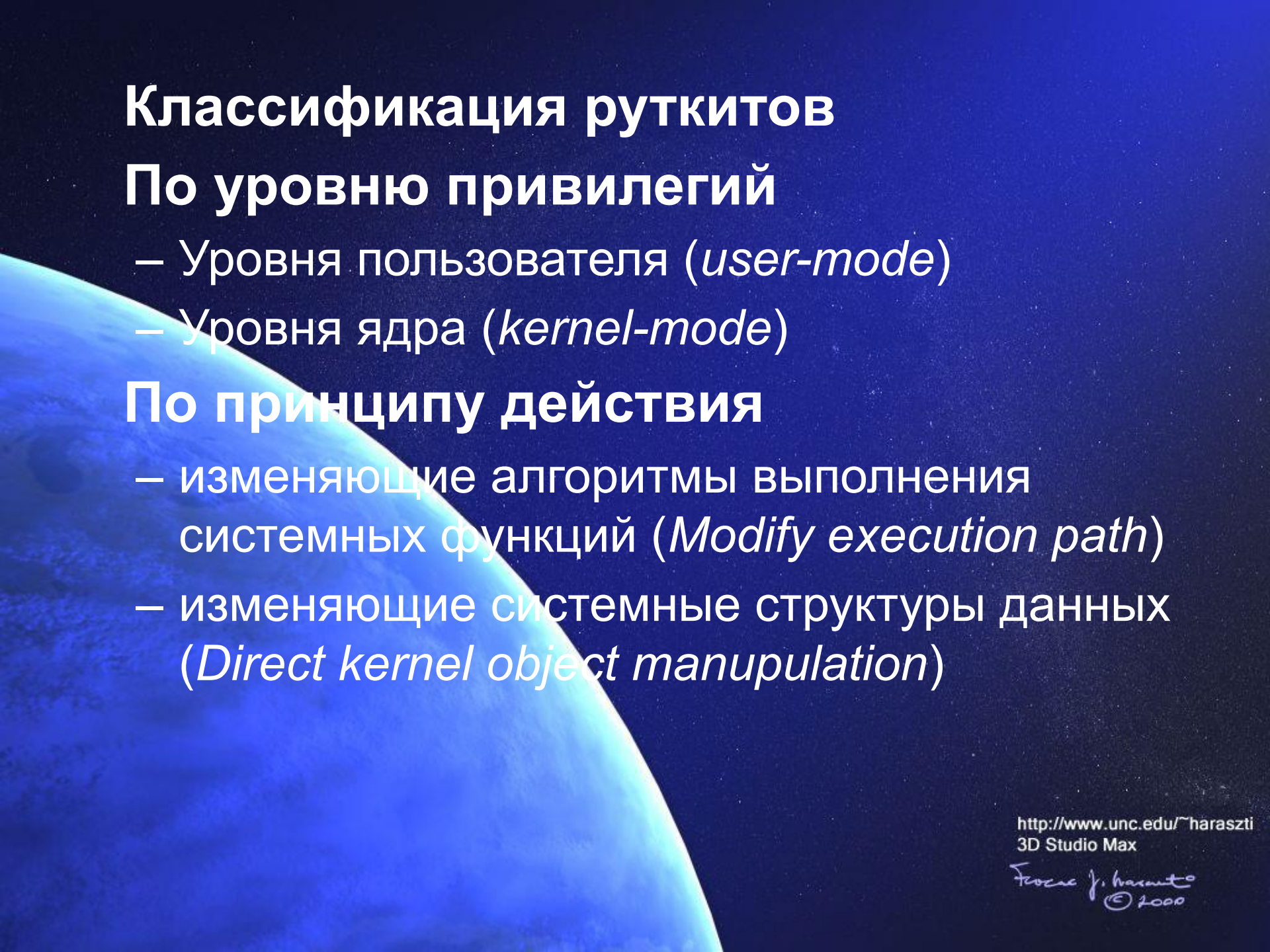
Классификация руткитов

По уровню привилегий

- Уровня пользователя (*user-mode*)
- Уровня ядра (*kernel-mode*)

По принципу действия

- изменяющие алгоритмы выполнения системных функций (*Modify execution path*)
- изменяющие системные структуры данных (*Direct kernel object manupulation*)



Основные методы реализации

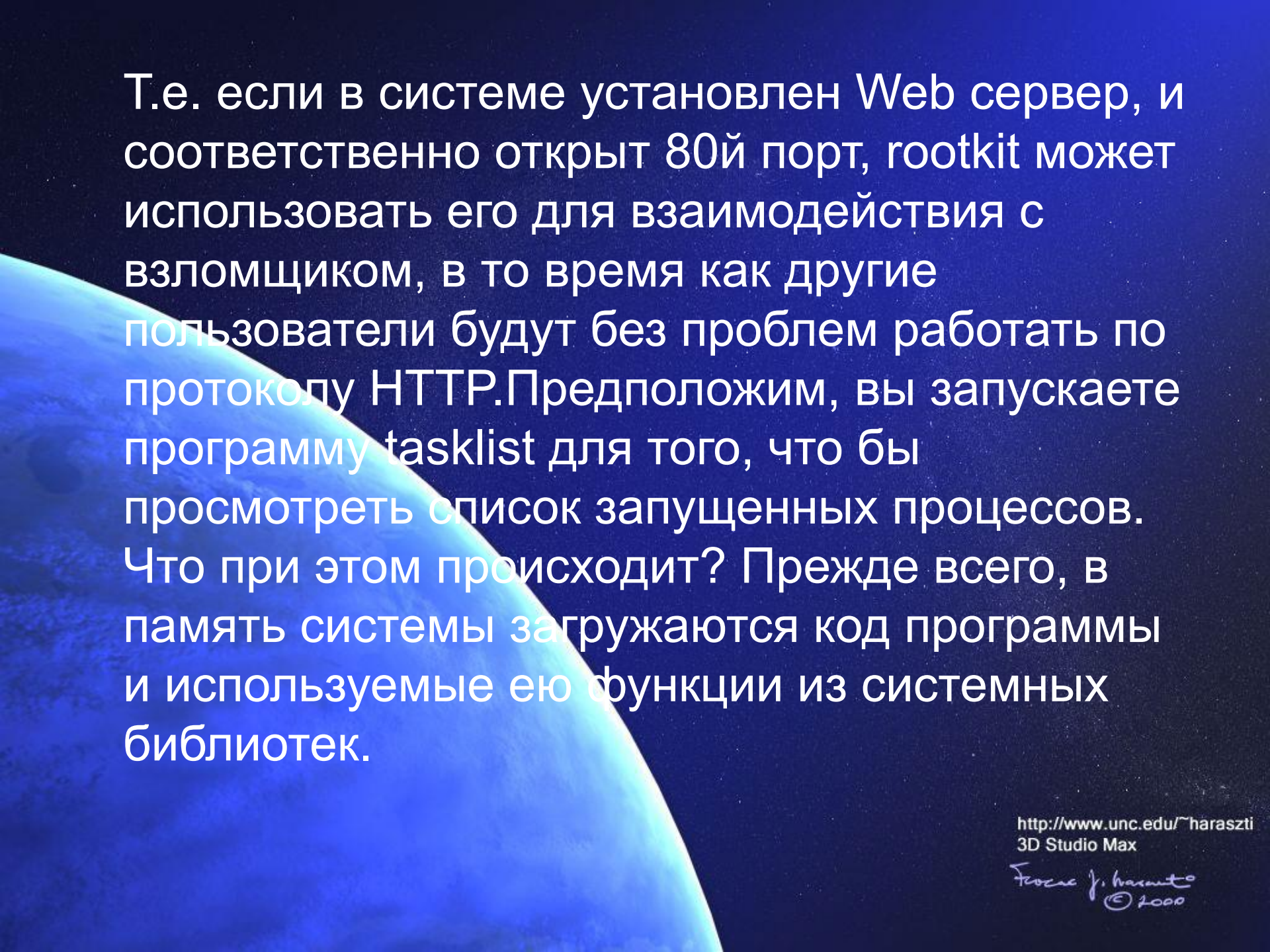
В UNIX

- реализуемые подменой основных системных утилит;
- реализованные в виде модуля ядра и основанные на патчинге VFS или перехвате таблицы системных вызовов (sys_call_table);
- основанные на модификации физической памяти ядра.

- **В Windows**
  В Windows из-за Windows File Protection переписывание системных файлов затруднено (хотя и его можно обойти!), поэтому основные способы внедрения в систему — модификация памяти.
- перехват системных функций Windows API (API hooking) на уровне пользователя
- то же на уровне ядра (перехват Native API)
- изменение системных структур данных

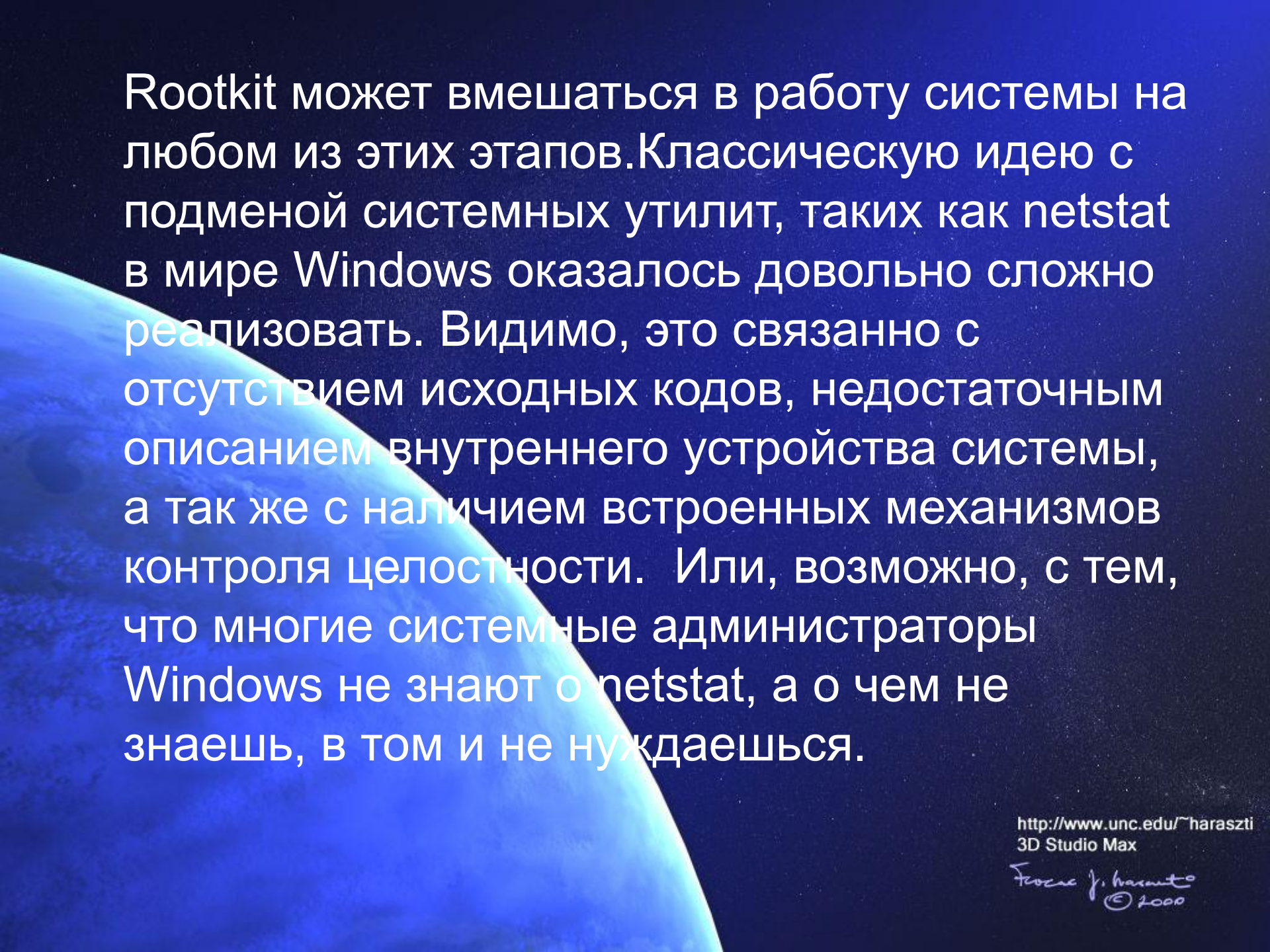
Большинство из реализаций современных rootkit могут прятать от пользователя файлы, папки и ключи реестра, скрывать запущенные программы, системные службы, драйверы и сетевые соединения. Т.е. злоумышленник имеет возможность создавать файлы и ключи реестра, запускать программы, работать с сетью и эта активность не будет обнаружена администратором. Кроме того, rootkits могут скрывать сетевую активность путем модификации стека протоколов TCP/IP. Так, например rootkit «Hacker Defender» перехватывает вызовы Winsock и может обрабатывать сетевой трафик до того как он будет передан приложению.

Т.е. если в системе установлен Web сервер, и соответственно открыт 80й порт, rootkit может использовать его для взаимодействия с взломщиком, в то время как другие пользователи будут без проблем работать по протоколу HTTP.Предположим, вы запускаете программу tasklist для того, что бы просмотреть список запущенных процессов. Что при этом происходит? Прежде всего, в память системы загружаются код программы и используемые ею функции из системных библиотек.



После этого программа вызывает функцию API, ответственную за выдачу списка процессов в системе (например, NtQuerySystemInformation из библиотеки Ntdll.dll). Функция NtQuerySystemInformation по сути заглушка, которая вызывает соответствующую функцию уровня ядра (ZwQuerySystemInformation), используя прерывание Windows (int 0x2E). Данная функция, работая уже в режиме ядра, напрямую обращается к памяти ядра и получает из структуры PsActiveProcessList список активных процессов, который отображается на экране.

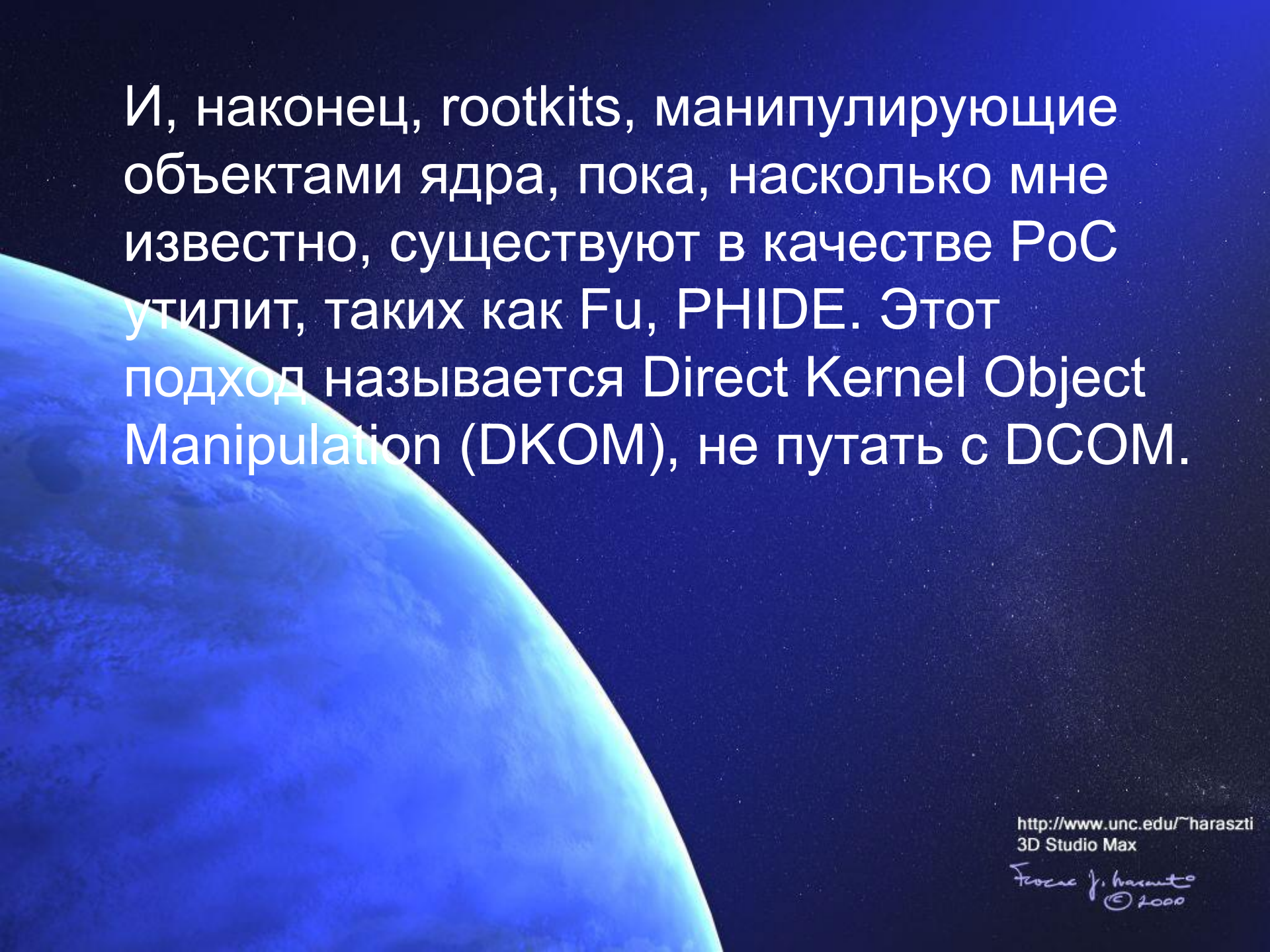
Rootkit может вмешаться в работу системы на любом из этих этапов.Классическую идею с подменой системных утилит, таких как netstat в мире Windows оказалось довольно сложно реализовать. Видимо, это связанно с отсутствием исходных кодов, недостаточным описанием внутреннего устройства системы, а так же с наличием встроенных механизмов контроля целостности. Или, возможно, с тем, что многие системные администраторы Windows не знают о netstat, а о чем не знаешь, в том и не нуждаешься.



Хотя некоторые отголоски этой идеи можно встретить и сейчас. Так, например, для запуска некоторых rootkit файл explorer.exe модифицируется таким образом, что бы считывать список автоматически исполняемых при загрузке программ из ключа отличного от HKCU[HKLM]\Software\Microsoft\Windows\CurrentVersion\Run. Естественно такая операция требует отключения системы Windows File Protection, но, имея права администратора в системе, это несложно осуществить даже без перезагрузки системы.В мире Windows в основе работы rootkits лежит модификация данных и кода программы в памяти операционной системы.

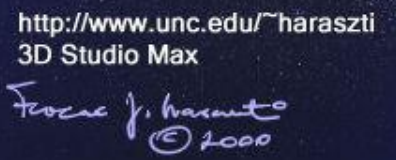
В зависимости от того, с какой областью памяти работает rootkit их можно подразделить на системы пользовательского уровня (User Level) и уровня ядра (Kernel Level, так же называемые KLT). В зависимости от метода реализации – основанные на модификации пути исполнения и манипулирующие только данными. Наиболее распространенными являются rootkit уровня пользователя, реализующие метод модификации пути исполнения. К подобным программам относятся Hacker Defender, Vanquish, AFX Rootkit. Менее распространены программы, модифицирующие пути исполнения на уровне ядра (например, He4Hook).

И, наконец, rootkits, манипулирующие объектами ядра, пока, насколько мне известно, существуют в качестве PoC утилит, таких как Fu, PHIDE. Этот подход называется Direct Kernel Object Manipulation (DKOM), не путать с DCOM.



Различные методы реализации Rootkit.

1.После запуска программа перечисляет все доступные для неё процессы в системе, после чего пытается перехватить определенные вызовы API. Перехват заключается в замене первых байт кода функции на безусловный переход к новому коду функции, предварительно сохраненному в адресном пространстве программы. Для этого выясняется адрес необходимой функции, затем в памяти программы отводится место под код нового варианта функции и её первоначального кода, который сохраняется для дальнейшего использования. Таким образом, когда пользовательская программа вызывает функцию API, например NtQuerySystemInformation, вызов передается функции предобработки данных, которая затем может вызвать исходную функцию, которая, в свою очередь, вернет результаты функции постобработки.

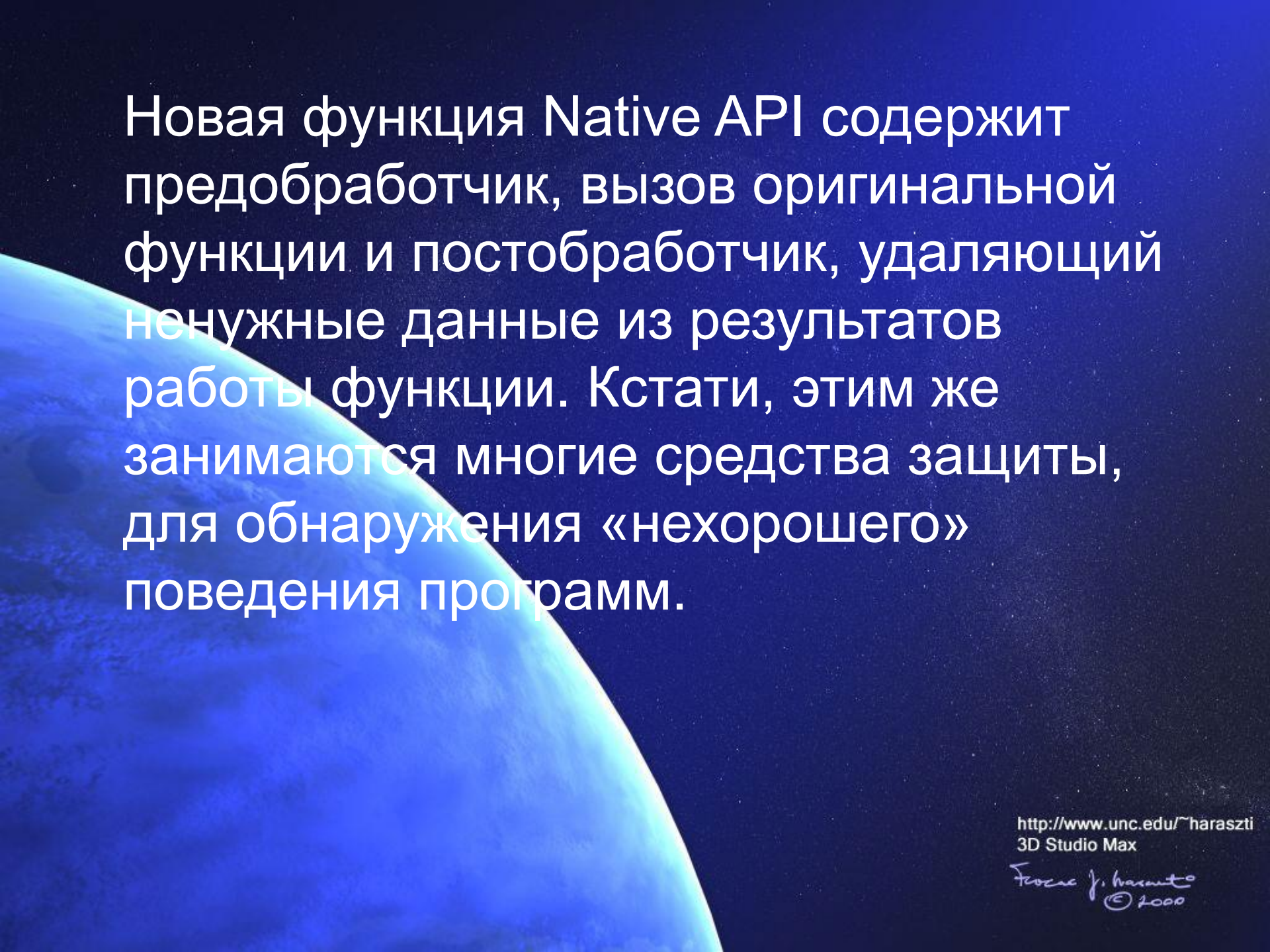


Функция постобработки модифицирует данные, которые вернула исходная функция, например, удаляя некоторые записи. Таким образом, программы, которые будут использовать перехваченные вызовы API, получат информацию не о реальном положении дел в системе, но уже обработанные Rootkit данные. Также Hacker Defender перехватывает функции запуска новых процессов, что позволяет ему заражать новые программы, запускаемые пользователем.

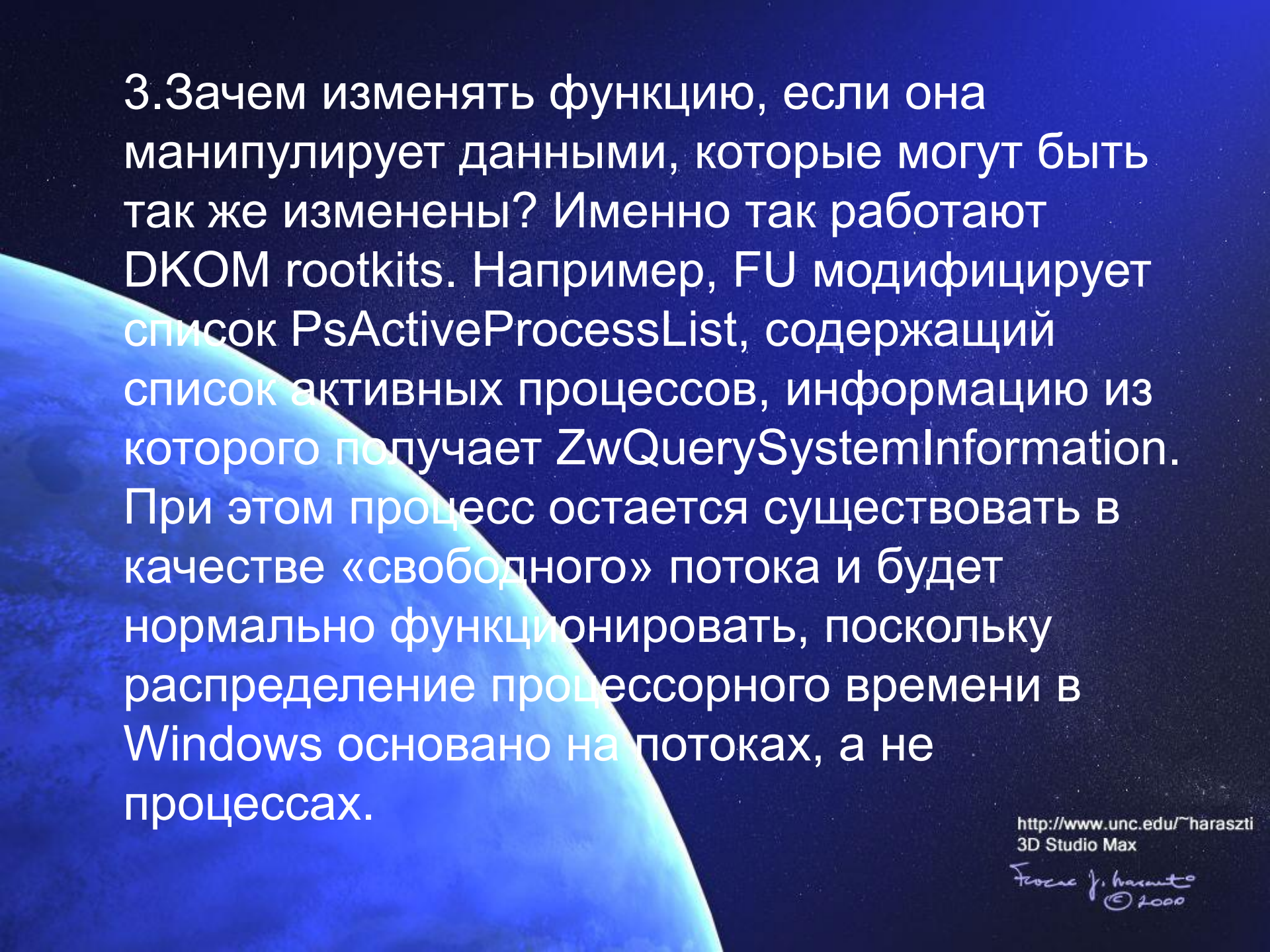
2.Идея модификации пути исполнения может быть реализована и на уровне ядра. При переходе в нулевое кольцо защиты функции API уровня пользователя вызывают прерывание 0x2E, которому в качестве параметра передается индекс необходимой функции. По данному индексу находится адрес соответствующего вызова API уровня ядра. Он хранится в структуре System Services Dispatch Table (SSDT) в виде упорядоченного списка адресов «низкоуровневых» функций (ZwCreateFile, ZwQuerySystemInformation и т.д.).

Эта таблица загружается в память операционной системы при старте микроядра ntoskrnl.exe. Некоторые rootkit модифицируют адрес, содержащийся в SSDT таким образом, чтобы он указывал на адрес обработчика, созданного им. Для этого может использоваться либо драйвер, либо user-land приложение, манипулирующее устройством \dev\physicalmemory, позволяющим (при наличии соответствующих полномочий, естественно) напрямую работать с памятью ядра.

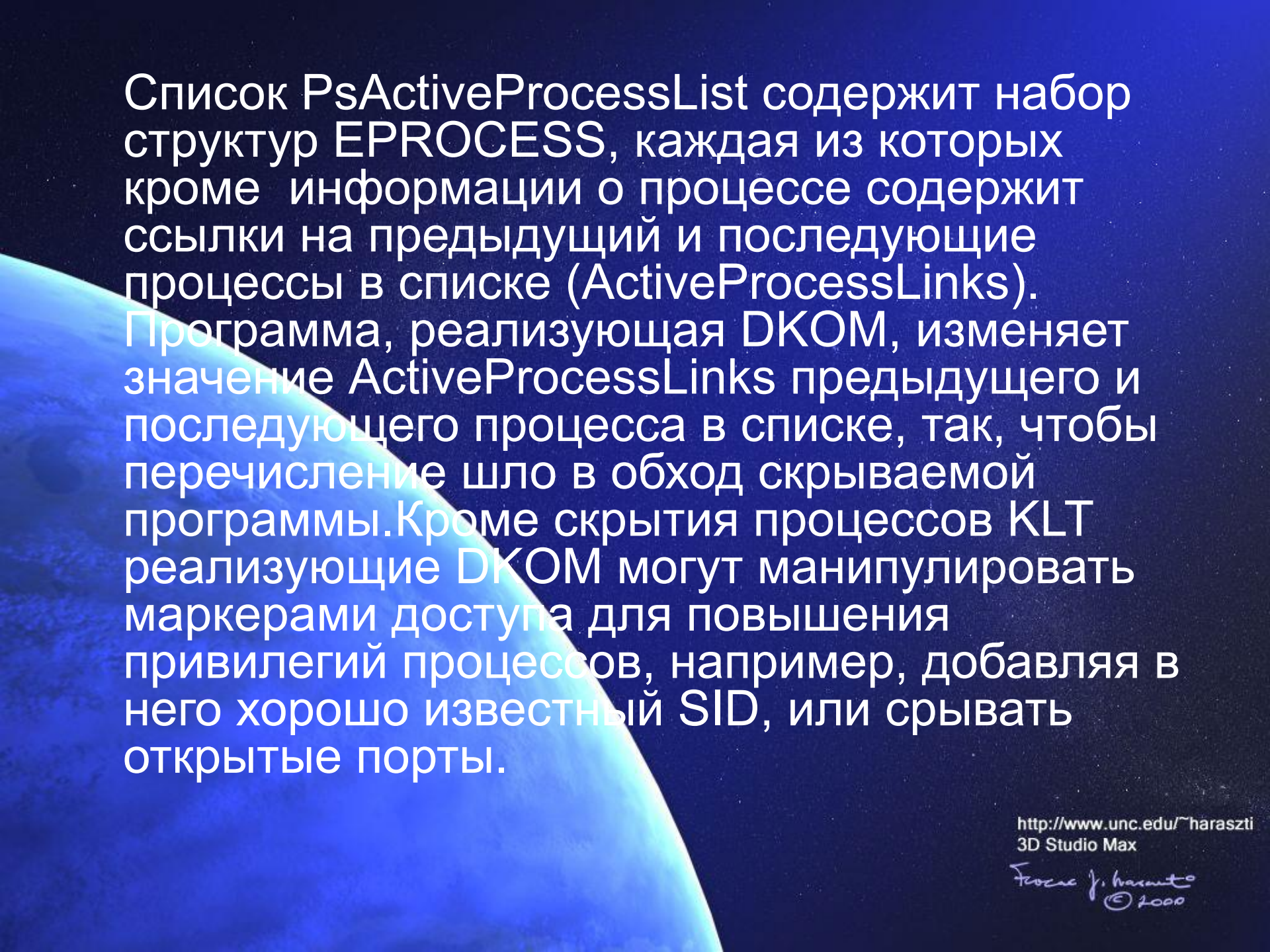
Новая функция Native API содержит предобработчик, вызов оригинальной функции и постобработчик, удаляющий ненужные данные из результатов работы функции. Кстати, этим же занимаются многие средства защиты, для обнаружения «нехорошего» поведения программ.



3.Зачем изменять функцию, если она манипулирует данными, которые могут быть так же изменены? Именно так работают DKOM rootkits. Например, FU модифицирует список PsActiveProcessList, содержащий список активных процессов, информацию из которого получает ZwQuerySystemInformation. При этом процесс остается существовать в качестве «свободного» потока и будет нормально функционировать, поскольку распределение процессорного времени в Windows основано на потоках, а не процессах.

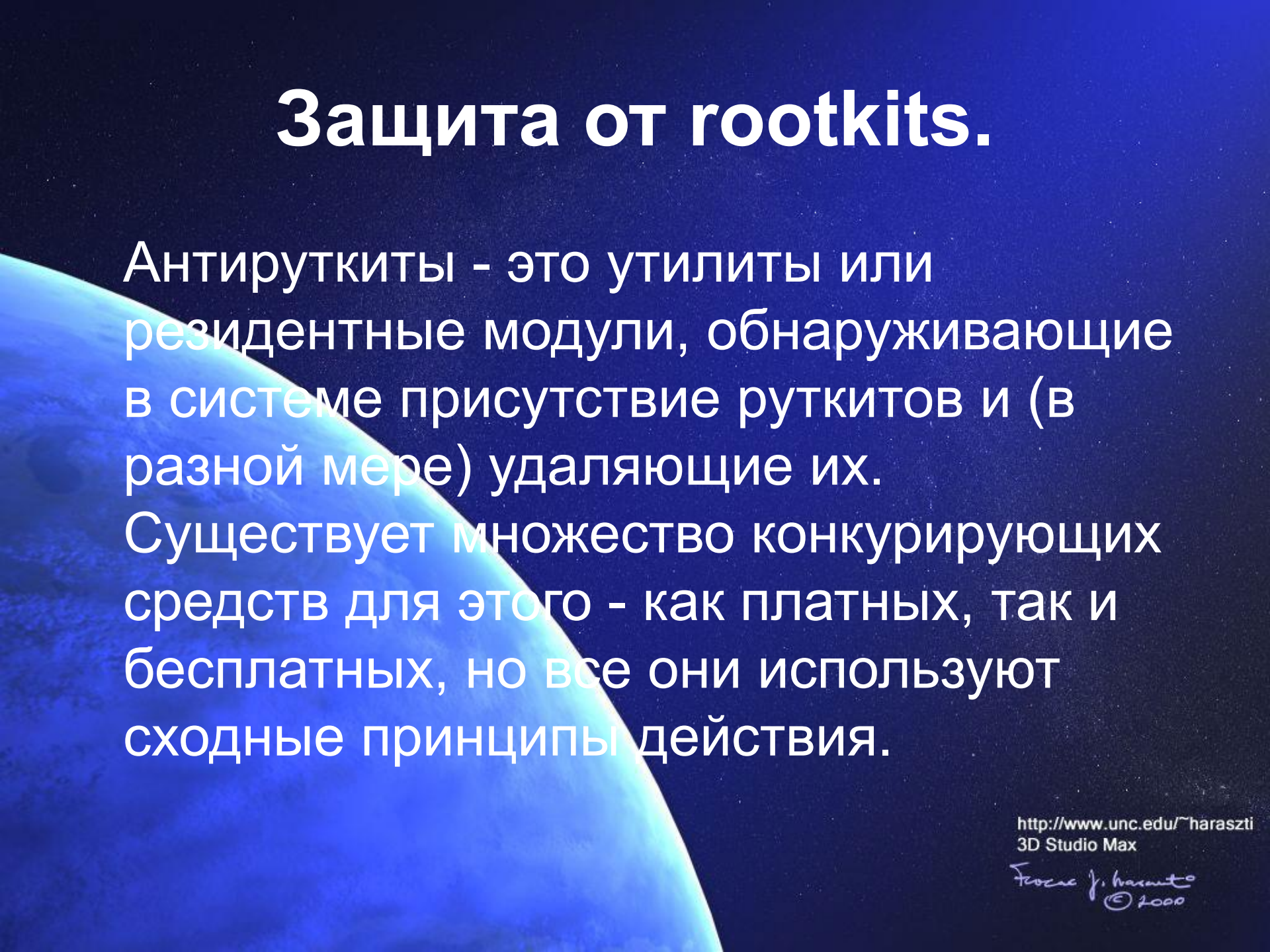


Список PsActiveProcessList содержит набор структур EPROCESS, каждая из которых кроме информации о процессе содержит ссылки на предыдущий и последующие процессы в списке (ActiveProcessLinks). Программа, реализующая DKOM, изменяет значение ActiveProcessLinks предыдущего и последующего процесса в списке, так, чтобы перечисление шло в обход скрываемой программы.Кроме скрытия процессов KLT реализующие DKOM могут манипулировать маркерами доступа для повышения привилегий процессов, например, добавляя в него хорошо известный SID, или срывать открытые порты.



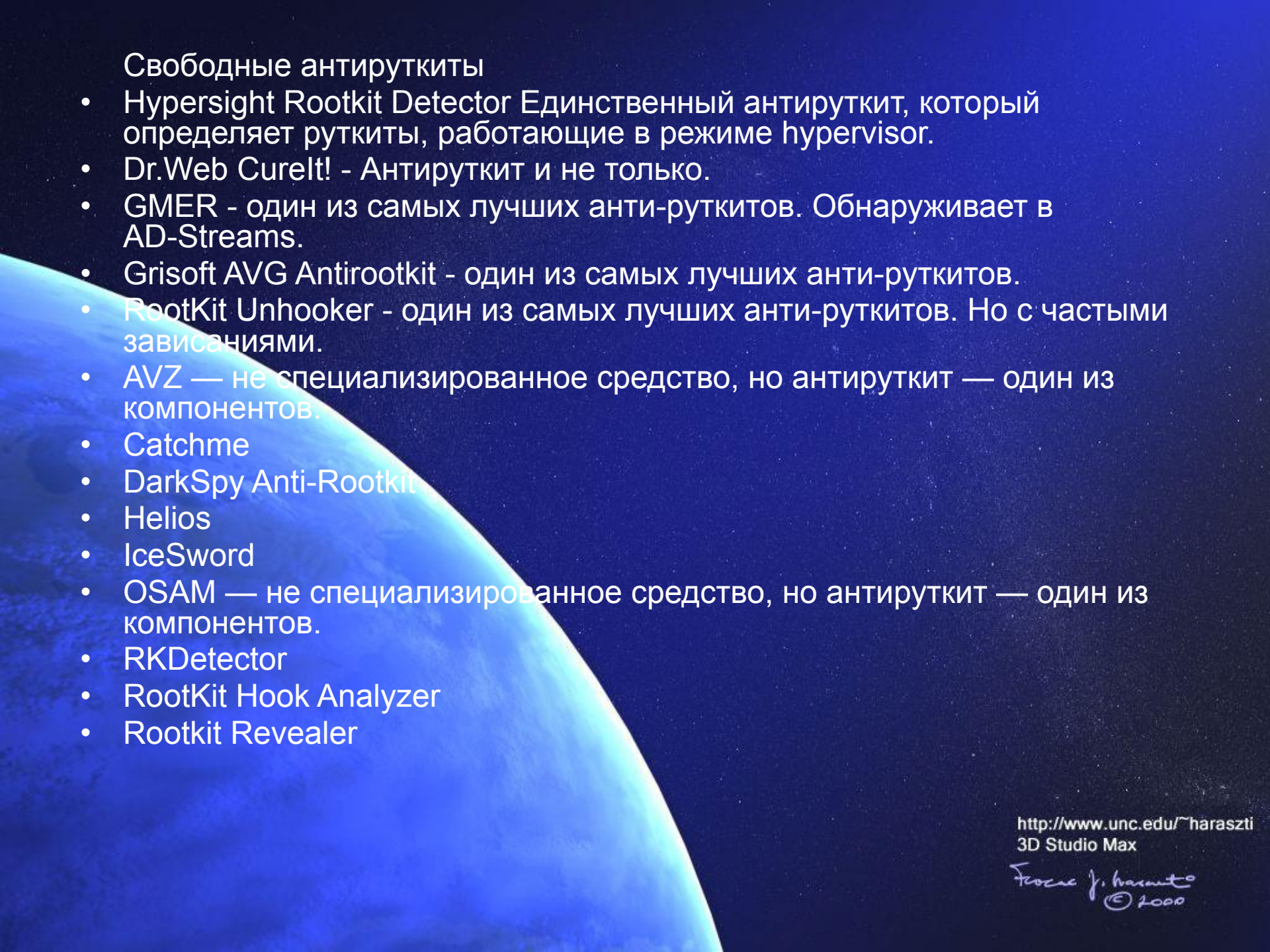
Защита от rootkits.

Антируткиты - это утилиты или резидентные модули, обнаруживающие в системе присутствие руткитов и (в разной мере) удаляющие их. Существует множество конкурирующих средств для этого - как платных, так и бесплатных, но все они используют сходные принципы действия.



Свободные антируткиты

- Hypersight Rootkit Detector Единственный антируткит, который определяет руткиты, работающие в режиме hypervisor.
- Dr.Web CureIt! - Антируткит и не только.
- GMER - один из самых лучших анти-руткитов. Обнаруживает в AD-Streams.
- Grisoft AVG Antirootkit - один из самых лучших анти-руткитов.
- RootKit Unhooker - один из самых лучших анти-руткитов. Но с частыми зависаниями.
- AVZ — не специализированное средство, но антируткит — один из компонентов.
- Catchme
- DarkSpy Anti-Rootkit
- Helios
- IceSword
- OSAM — не специализированное средство, но антируткит — один из компонентов.
- RKDetector
- RootKit Hook Analyzer
- Rootkit Revealer



Коммерческие антируткиты

- Avira Antivir Rootkit
- BitDefender Antirootkit
- F-Secure BackLite
- McAfee Rootkit Detective
- Panda AntiRootkit
- Sophos Anti-Rootkit
- Trend Micro RootkitBuster
- Kaspersky: AntiVirus и Internet Security - комплексные защиты, включающие в себя антируткиты